



**Решения заданий заключительного тура
олимпиады школьников «Гранит науки»
по профилю Информатика
в 2020/2021 учебном году**



ИНФОРМАТИКА
БИЛЕТ 1. ВАРИАНТ 1

1. С тех пор как в жизни Васи появились электронные таблицы, он непрерывно в них что-то считает. Вот и на каникулах он составил таблицу, показанную на рисунке ниже.

	Q	R	S	T	U	V
17		1				
18	1					
19						
20						
21						
22						

В ячейке S17 записана формула $=\$Q18+R17$ и распространена по строке вправо. В ячейке Q19 записана формула $=Q18+СЧЁТЗ(ДВССЫЛ("Q18:Q18"))$ и распространена по столбцу вниз. В ячейке R18 записана формула $=СТЕПЕНЬ(\$Q18;2)+R\$17^{\text{LOG}(100)}$ и с ее помощью заполнена вся таблица. Какое значение получит Вася в ячейке V22, если в ней записана формула $\{=\text{МУМНОЖ}(R17:U17;Q18:Q21)-\text{СУММЕСЛИ}(Q18:Q21;">3";R18:U21)\}$? Дайте пояснения. Укажите значения произведения матриц и суммы элементов. **(8 баллов)**

Ответ: 13

Решение:

	1	2	3	4	
1	2	5	10	17	
2	5	8	13	20	
3	10	13	18	25	
4	17	20	25	32	
					13

Произведение матриц = 30. Сумма элементов равна 17.

2. Два гнома решили сыграть в игру «20 драгоценных камней». Они выложили в ряд 20 драгоценных камней. По условию игры можно поочередно брать 1, 2 или 3 камня за раз. Можно брать только **смежные** камни, лежащие рядом. Брать одновременно камни, лежащие по разную сторону от пустого места (камень или камни между ними забрали в предыдущий ход), нельзя. Пропускать свой ход нельзя. **Выигрывает** тот, кто заберет **последние** камни или камень. Напишите алгоритм **выигрышной стратегии** для гнома, ходящего **первым**. Поясните алгоритм графически. Напишите обобщенный алгоритм для любого четного количества камней. Укажите нижнюю границу применимости алгоритма. **(10 баллов)**

Решение:

Используется Стратегия_1 (смежные предметы). Всегда выигрывает Игрок_1 (игрок, делающий ход первым). **Суть Стратегия_1:**

Ход_1: Игрок_1 берет 2 предмета из центра ряда (в данном случае 10-й и 11-й предмет). Получается две несмежные половины с равным количеством предметов.

Ход_2: Игрок_2 (игрок, делающий ход вторым) берет любое разрешенное количество предметов из любой половины (но только из одной, т.к. можно брать только смежные предметы).

Ход_3: Игрок_1 повторяет Ход_2 зеркально в другой половине.

Ход_4: равноценен Ходу_2.

Ход_5: зеркальное отражение Хода_4.

И так далее до **Победа** Игрок_1. Проиграть невозможно.



Иллюстрация на рисунке ниже. Алгоритм подходит для **любого четного количества** предметов.

Алгоритм для Игрок_1 (обобщенный):

Шаг 1 Взять $j=N/2$ и $j=(N/2+1)$ предметы.

Шаг 2. Затем отзеркалить противника по формуле: Взять_предмет_ j , $j=N-i+1$, где N – общее число предметов; i – номер_предмета, взятого Игрок_2.

Если Остаток = 0 – значит СТОП_Выиграл.

Нижняя граница – 6 предметов.

Вариант 1 Стратегия 1 Победитель Игрок 1	Игрок_1 - ходит первым										Игрок_2 - ходит вторым										Можно брать 1, 2 или 3 СМЕЖНЫХ предмета. Всего N предметов, N четное.
Исходное состояние	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
Ход 1 - забрал 10 и 11	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
	Осталось 9 предметов										=	Осталось 9 предметов									
Ход 2 - любой набор из 1,2 или 3-х предметов в любой половине (смежные)	1	2	3	4	5	6	7	8	9			12	13	14	15	16	17	18	19	20	
Ход 3 - зеркальное отражение хода 2	1			4	5	6	7	8	9			12	13	14	15	16	17	18	19	20	
	1					6								6						1	
Ход 4 - любой набор из 1,2 или 3-х предметов в любой половине (смежные)	1			4	5	6	7	8	9			12	13	14	15	16	17			20	
	1					6						1			2					1	
Ход 5 - зеркальное отражение хода 4	1			4	5	6	7	8	9			12				16	17			20	
	1			2					1			1			2					1	
Ход 6 - любой набор из 1,2 или 3-х предметов в любой половине (смежные)	1			4	5				9			12				16	17			20	
	1			2					1			1			2					1	
	И так далее до конца. В конце у Игрок_1 остается разрешенное количество смежных предметов.																				

3. На станции «Сортировка» специальный семафор для маневрового тепловоза состоит из 4 секций, расположенных буквой Т (горизонтальная составляющая содержит три секции, а вертикальная две). Каждая секция может гореть синим светом или быть выключенной. На работу семафора наложены следующие ограничения: а) должно гореть не менее двух секций; б) не может гореть только две секции в горизонтальном ряду. Нарисуйте разрешенные комбинации семафора. Составьте таблицу истинности работы семафора. Запишите логическую функцию, описывающую разрешенные комбинации семафора, в дизъюнктивной нормальной форме. Упростите полученное выражение, оставаясь в базисе И, ИЛИ, НЕ. (12 баллов)



Решение:

	Запрещенные комбинации				Разрешенные комбинации		
1				9			
2				10			
3				11			
4				12			
5				13			
6				14			
7				15			
8				16			

Обозначим буквами секции семафора: a, b, c – слева направо верхний ряд, d – секция в нижнем ряду. Составим таблицу истинности функции F .

N	a	b	c	d	F
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	0
7	0	1	1	1	1
8	1	0	0	0	0
9	1	0	0	1	1
10	1	0	1	0	0
11	1	0	1	1	1
12	1	1	0	0	0
13	1	1	0	1	1
14	1	1	1	0	1
15	1	1	1	1	1

Функция F истинна (равна 1) на номерах наборов 3,5,7,9,11,13,14 и 15.

$$F = (\bar{a}\bar{b}lсld) \vee (\bar{a}lb\bar{c}ld) \vee (\bar{a}lbлсld) \vee (a\bar{b}\bar{c}ld) \vee (a\bar{b}лсld) \vee (a\bar{b}л\bar{c}ld) \vee (a\bar{b}лсld) \vee (a\bar{b}лсld)$$

Упростим функцию. По ТИ видно, что $сld$ всегда дают $F=1$. Запишем новое выражение:

$$F = (сld) \vee (\bar{a}lb\bar{c}ld) \vee (a\bar{b}\bar{c}ld) \vee (a\bar{b}л\bar{c}ld) \vee (a\bar{b}лсld)$$



<i>N</i>	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>F</i>
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	0
7	0	1	1	1	1
8	1	0	0	0	0
9	1	0	0	1	1
10	1	0	1	0	0
11	1	0	1	1	1
12	1	1	0	0	0
13	1	1	0	1	1
14	1	1	1	0	1
15	1	1	1	1	1

В полученном выражении можно минимизировать выражения в 3 и 4 скобках:

$$(a\bar{b}\bar{c}ld)\vee(a\bar{b}c\bar{c}ld)=(a\bar{c}ld).$$

Итоговое выражение в базисе И, ИЛИ, НЕ: $F = (cld)\vee(\bar{a}b\bar{c}ld)\vee(a\bar{c}ld)\vee(a\bar{b}c\bar{c}ld)$

4. Иван устроился работать системным администратором в филиал НИИ «Технологии». В первый же день работы Иван получил ответственное задание. Отдел информационных технологий НИИ «Технологии» использует диапазон IP-адресов 192.168.160.0/22, в котором настроено адресное пространство сетей филиалов из 6 подсетей. В результате анализа настроек были установлены адреса компьютеров сетевых администраторов каждой подсети. Известно, что адреса этих компьютеров являются первыми возможными адресами узлов для каждой подсети. Адресное пространство при этом использовано максимально эффективно.

У Ивана имеются следующие познания:

- одному устройству должен соответствовать один IP-адрес вида XXX.XXX.XXX.XXX, при этом XXX – число в диапазоне [0:255], называемое октетом;
- устройства внутри одной подсети должны беспрепятственно взаимодействовать друг с другом и должны группироваться по их расположению;
- выделенный размер подсети содержит минимально необходимое количество адресов;
- подсети большего размера должны находиться в начале адресного диапазона сети;
- в каждом сегменте сети первый адрес зарезервирован под идентификатор подсети, а последний – под широковещательные (broadcast) запросы;
- выделенный размер подсети записывается как 2^n-2 , где n – минимальное количество бит, которое может содержать все номера адресов подсети.



Название подсети	Адрес ПК сетевого администратора	Выделенный размер	Адрес сети	Маска с префиксом	Десятичная маска	Диапазон доступных адресов	Широковещание
А	192.168.160.1						
В	192.168.162.1						
С	192.168.163.1						
Администрация	192.168.163.129						
Пост охраны	192.168.163.161						
IT-отдел	192.168.163.177						

Заполнение информации вызвало у системного администратора затруднения. Необходимо помочь Ивану заполнить таблицу недостающими данными. **(20 баллов)**

Решение:

Название подсети	Адрес ПК сетевого администратора	Выделенный размер	Адрес сети	Маска с префиксом	Десятичная маска	Диапазон доступных адресов	Широковещание
А	192.168.160.1	510	192.168.160.0	/23	255.255.254.0	192.168.160.1 - 192.168.161.254	192.168.161.255
В	192.168.162.1	254	192.168.162.0	/24	255.255.255.0	192.168.162.1 - 192.168.162.254	192.168.162.255
С	192.168.163.1	126	192.168.163.0	/25	255.255.255.128	192.168.163.1 - 192.168.163.126	192.168.163.127
Д	192.168.163.129	30	192.168.163.128	/27	255.255.255.224	192.168.163.129 - 192.168.163.158	192.168.163.159
Е	192.168.163.161	14	192.168.163.160	/28	255.255.255.240	192.168.163.161 - 192.168.163.174	192.168.163.175
Ф	192.168.163.177	2	192.168.163.176	/30	255.255.255.252	192.168.163.177 - 192.168.163.178	192.168.163.179

5. Администратору сообществ ВК Николаю необходимо определить, какие участники состоят в сообществе «Кубок веломарафонов Ленинградской области» (3353 участника) и одновременно в сообществе «Грибы Ленинградской области» (74656 участников), и вывести на экран получившийся список. У администратора есть два упорядоченных списка участников каждой группы. В каждом списке участник представлен одним и тем же восьмизначным цифровым кодом. Без повторений внутри списка. Списки упорядочены по возрастанию кодов. Помогите Николаю составить алгоритм и программу для решения этой задачи, считая, что списки пригодны для непосредственной обработки выбранным языком программирования и уже загружены в память. Правильность решения докажете на простом примере. В решении **нельзя** использовать встроенные в язык операции, методы или свойства для работы со структурами данных, кроме как для определения длины структуры или добавления (удаления) элементов структуры. Код необходимо снабдить комментариями. **(25 баллов)**

Решение:

Алгоритм решения задачи:

Задача на пересечение множеств. Пересечение двух упорядоченных множеств **А** и **В** – это множество только с теми элементами **А** и **В**, которые одновременно принадлежат обоим



множествам, без дублей. Сложность алгоритма $O(m+n)$, где m и n – длины входных множеств A и B соответственно.

Необходимо написать программу, аргументы которой – два упорядоченных массива для пересечения. Программа возвращает третий массив, содержащий элементы, входящие одновременно в оба массива.

Шаг 1. Необходимо взять два исходных массива и пустой третий.

Шаг 2. Создать и обнулить три счетчика индексов массивов, например: $i=0, j=0, k=0$.

Шаг 3. Сравнить верхние (нулевые) элементы массивов.

Если $M_{1i} = M_{2j}$, то: $M_{3k} = M_{1i}, i=i+1, j=j+1, k=k+1$.

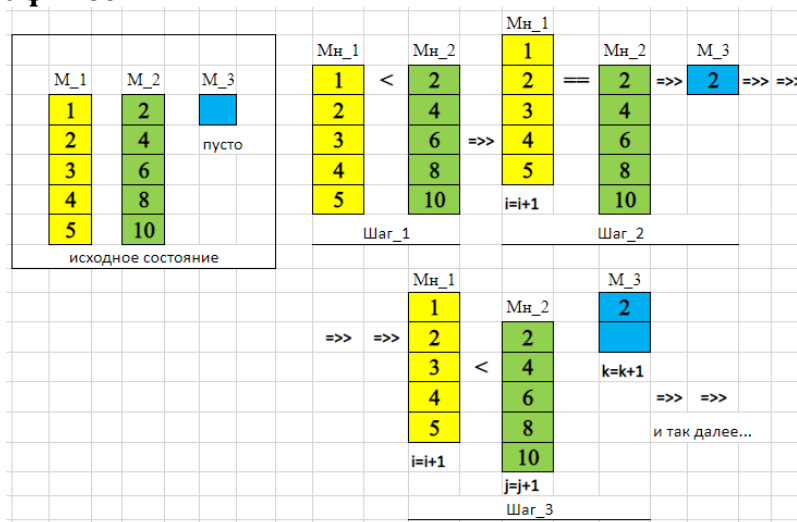
Если $M_{1i} < M_{2j}$, то: $i=i+1$ – поднимаем M_1 вверх на один элемент.

Если (иначе, т.к. вариантов больше нет) $M_{1i} > M_{2j}$, то: $j=j+1$ – поднимаем M_2 вверх на один элемент.

Шаг 4. И так в цикле while пока $i < L_1$ и $j < L_2$, где L – длина массива.

Проверить индексацию массивов и правильность определения верхней границы массива в условии цикла while.

Алгоритм графически:



Пример решения в MathCad:

Задача №5 - Пересечение множеств

$$M_1 := \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{pmatrix} \quad M_2 := \begin{pmatrix} 2 \\ 4 \\ 6 \\ 8 \\ 10 \end{pmatrix} \quad \begin{array}{l} \text{length}(M_1) = 5 \\ \text{length}(M_2) = 5 \end{array}$$

```

intersec_sort_arr(a,b) := (aL ← length(a) bL ← length(b)
(i ← 0 j ← 0 k ← 0)
while (i < aL) ∧ (j < bL)
  if ai = bj
    ck ← ai
    (i ← i + 1 j ← j + 1 k ← k + 1)
  otherwise
    if ai < bj
      i ← i + 1
    otherwise
      j ← j + 1
c

```

$$\text{intersec_sort_arr}(M_1, M_2) = \begin{pmatrix} 2 \\ 4 \end{pmatrix}$$

Пример решения на Python:

```
Гранит_2021
File Edit Format Run Options Window Help

1 ##Задача №5 - Пересечение множеств
2 ##Дано два массива в виде списков
3 aList=[1,2,3,4,5]
4 bList=[2,4,6,8,10]
5 cList=[]##Пустой массив
6 len(aList) - длина массива
7 i=0##Счетчики
8 j=0
9 k=0
10 while i<len(aList) and j<len(bList):
11     if aList[i]==bList[j]:
12         cList.append(aList[i])##Добавление элемента в конец списка
13         i+=1##Увеличение счетчика на 1
14         j+=1
15         k+=1
16     elif aList[i]<bList[j]:
17         i+=1
18     else:
19         j+=1
20 print(cList)
Ln: 9 Col: 3

Python 3.8.2 Shell
File Edit Shell Debug Options Window Help

Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit
(Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: C:/Users/User/Downloads/Гранит_2021_2_этап_Косарев_Задача_5.py
====
[2, 4]
Ln: 6 Col: 4
```

Решение PascalABC.NET:

```
PascalABC.NET 3.7.2
Файл Правка Вид Программа Сервис Модули Помощь

•Гранит_2021
1 program Zadacha_5;//Необязательный заголовок
2 //Задача на пересечение множеств
3 //В языке есть встроенные множества "set"
4 //и метод a.Intersect(b) пересечения множеств
5 //По условиям задачи так делать нельзя и метод
6 //выдает последовательность, а не массив.
7
8 //Динамические массивы с данными для сортировки
9 var
10 a: array of integer := (1, 2, 3, 4, 5);
11 b: array of integer := (2, 4, 6, 8, 10);
12 //Выходной массив. изначально пустой
13 c: array of integer;
14 i, j, k: integer;//Счётчики=0 по умолчанию
15
16 begin
17 while (i < a.Length) and (j < b.Length) do
18     if a[i] = b[j] then
19         begin//Ветка true
20             SetLength(c, k + 1);
21             c[k] := a[i];
22             i := i + 1; j := j + 1; k := k + 1;
23         end
24     else//Ветка false
25         if a[i] < b[j] then i := i + 1
26         else j := j + 1;
27     writeln(c);
28
29 //А теперь с помощью встроенного метода
30 a.Intersect(b).Println;
31 end.

Окно вывода
[2, 4]
2 4
Ln: 1 Col: 1
```


Пример решения на VBA:

```
Гранит Науки = 2(1).xslm - Module1 (Code)
(Zadacha_2)

Sub Zadacha_2 ()

    Set A = Cells(1, 1).CurrentRegion      'Множество A
    Set B = Cells(1, 2).CurrentRegion      'Множество B
    L_A = A.Rows.Count                    'Количество элементов множества A
    L_B = B.Rows.Count                    'Количество элементов множества B

    Dim i As Integer                      'Счетчик A
    Dim j As Integer                      'Счетчик B
    Dim k As Integer                      'Счетчик C

    i = 1: j = 1: k = 1 'Инициализация счетчиков

    While (i <= L_A) And (j <= L_B)      'Условие завершения программы = окончание перебора
                                        'элементов множества A и множества B
        M_1 = Cells(i, 1)                'Обращение к текущему элементу множества A
        M_2 = Cells(j, 2)                'Обращение к текущему элементу множества B

        If M_1 = M_2 Then                 'Текущие элементы множеств A и B пересеклись
            M_3 = M_1: Cells(i, 3) = M_3  'Результат промешаем в множество C

            i = i + 1: j = j + 1: k = k + 1 'Значение счетчиков увеличиваем
        Else:
            If M_1 < M_2 Then i = i + 1 Else: j = j + 1 'Перемещение по множеству A либо B

        End If
    Wend
End Sub
```

6. Сельскохозяйственный дрон должен выполнить фотосъемку поля, засеянного пшеницей. Поле разделено на секции размером 1680x640 м. Фотосенсор дрона имеет квадратную форму. Реальный размер квадратного фотографируемого участка секции поля может изменяться за счет высоты зависания дрона. Дрону необходимо сфотографировать весь участок без пропусков и перекрытий минимальным количеством снимков. Напишите алгоритм и универсальную программу, которая на основе размеров прямоугольного участка вычисляет и выводит в качестве ответа размер стороны квадрата, который необходимо сфотографировать дрону. Правильность решения докажете поэтапно вручную по коду. Код необходимо снабдить комментариями. **(25 баллов)**

Решение:

Алгоритм решения задачи

Задача на использование рекурсии и алгоритма «Разделяй и властвуй». Есть некоторая поверхность прямоугольной формы с заданными сторонами. Ее необходимо разбить на минимальное количество одинаковых квадратов. Соответственно площадь квадратов должна быть максимальной. Решается с помощью «Алгоритма Евклида». Суть алгоритма: определить базовый случай и применить рекурсию. Базовый случай: соотношение сторон участка 2:1. Графическая иллюстрация ниже. Ответ: сторона квадрата = 80 м. Проверка поэтапно на рисунке с кодом. Проверка по сторонам: 1680:80=21; 640:80=8.

Пример решения на PascalABC.NET:

```
1 program Zadacha_6; //Необязательный заголовок
2 //Задача на определение максимального квадрата,
3 //которым можно заполнить прямоугольник.
4 var //Переменные для хранения сторон прямоугольника
5 //a,b - стороны участка, m - остаток от деления mod
6 m: integer;
7 //рекурсивная функция
8 function kvadrat(a,b: integer):integer;
9 var //аккумулятор для перемены местами a и b
10 akkum: integer;
11 begin //если указали стороны в неправильном порядке
12 if a<b then
13 begin //меняем местами a и b
14 akkum:=a;
15 a:=b;
16 b:=akkum;
17 end;
18 println('a=',a); //для контроля хода решения
19 println('b=',b);
20 //базовый случай, когда отношение сторон 2/1
21 if (a mod b)=0 then //это и есть базовый случай
22 begin
23 //kvadrat:=b; //можно и так и как ниже
24 Result:=b; //Определяем что возвращает функция
25 Println('Сторона квадрата = ', Result);
26 exit
27 end
28 else //базовый случай не достигнут
29 begin
30 m:=a mod b;
31 //a:=b;
32 //b:=m;
33 kvadrat(b,m); //запускаем рекурсию
34 end;
35 end;
36
37 begin //основная программа
38 kvadrat(1680,640); //вызов рекурсивной функции
39 end.
```

Окно вывода

```
a= 1680
b= 640
a= 640
b= 400
a= 400
b= 240
a= 240
b= 160
a= 160
b= 80
Сторона квадрата = 80
```

Окно вывода Список ош



ИНФОРМАТИКА
БИЛЕТ 1. ВАРИАНТ 2

1. С тех пор как в жизни Васи появились электронные таблицы, он непрерывно в них что-то считает. Вот и на каникулах он составил таблицу, показанную на рисунке ниже.

	Q	R	S	T	U	V
17		-1				
18	-1					
19						
20						
21						
22						

В ячейке S17 записана формула $=\$Q18+R17$ и распространена по строке вправо. В ячейке Q19 записана формула $=Q18+СЧЁТЗ(ДВССЫЛ("Q\$18:\$Q18"))$ и распространена по столбцу вниз. В ячейке R18 записана формула $=СТЕПЕНЬ(\$Q18;2)+R\$17^{\text{LOG}(100)}$ и с ее помощью заполнена вся таблица. Какое значение получит Вася в ячейке V22, если в ней записана формула $\{=МУМНОЖ(R17:U17;Q18:Q21)-СУММЕСЛИ(Q18:Q21;">2";R18:U21)\}$? Дайте пояснения. Укажите значения произведения матриц и суммы элементов. **(8 баллов)**

Ответ: -10

Решение:

	-1	-2	-3	-4
-1	2	5	10	17
0	1	4	9	16
1	2	5	10	17
2	5	8	13	20

-10

Произведение матриц = -10. Сумма элементов равна 0.

2. Два гнома решили сыграть в игру «21 драгоценный камень». Они выложили в ряд **21** драгоценный камень. По условию игры можно поочередно брать 1, 2 или 3 камня за раз. Можно брать только **смежные** камни, лежащие рядом. Брать одновременно камни, лежащие по разную сторону от пустого места (камень или камни между ними забрали в предыдущий ход), нельзя. Пропускать свой ход нельзя. Выиграет тот, кто заберет последние камни или камень. Напишите алгоритм **выигрышной стратегии** для гнома, ходящего **первым**. Поясните алгоритм графически. Напишите обобщенный алгоритм для любого нечетного количества камней. Укажите нижнюю границу применимости алгоритма. **(10 баллов)**

Решение:

Используется Стратегия_1 (смежные предметы). Всегда выигрывает Игрок_1 (игрок, делающий ход первым). **Суть Стратегия_1:**

Ход_1: Игрок_1 берет **3** предмета из центра ряда (в данном случае 10, 11-й и 12-й предмет). Получается две несмежные половины с равным количеством предметов.

Ход_2: Игрок_2 (игрок, делающий ход вторым) берет любое разрешенное количество предметов из любой половины (но только из одной, т.к. можно брать только смежные предметы).

Ход_3: Игрок_1 повторяет Ход_2 зеркально в другой половине.

Ход_4: равноценен Ходу_2.

Ход_5: зеркальное отражение Хода_4.

И так далее до **Победа** Игрок_1.



Алгоритм подходит для **любого НЕчетного количества** предметов.

Алгоритм для Игрок_1:

Шаг 1 - взять $j=(N-1)/2$, $((N-1)/2 +1)$ и $((N-1)/2 +2)$ предметы.

Затем отзеркалить противника по формуле: Взять_предмет_ j , $j=N-i+1$, где N - общее число предметов; i - номер предмета взятого Игрок_2.

Если Остаток = 0 - значит СТОП_Выиграл.

Нижняя граница - 7 предметов.

	Игрок_1 - ходит первым												Можно брать 1, 2 или 3 СМЕЖНЫХ предмета. Всего N предметов, N нечетное.								
Вариант 2 Стратегия 1	Игрок_2 - ходит вторым																				
Победитель Игрок 1																					
Исходное состояние	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Ход 1 - забрал 10, 11 и 12	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
	Осталось 9 предметов									=	Осталось 9 предметов										
Ход 2 - любой набор из 1,2 или 3-х предметов в любой половине (смежные)	1	2	3	4	5	6	7	8	9				13	14	15	16	17	18	19	20	21
	Далее игра сводится к Варианту1																				
Ход 3 - зеркальное отражение хода 2	1			4	5	6	7	8	9				13	14	15	16	17	18	19	20	21
	1	6						=	6						1						
	И так далее до конца. В конце у Игрок_1 остается разрешенное количество смежных предметов.																				

3. На станции «Сортировка» специальный семафор для маневрового тепловоза состоит из 4 секций, расположенных буквой Т (горизонтальная составляющая содержит три секции, а вертикальная две). Каждая секция может гореть синим светом или быть выключенной. На работу семафора наложены следующие ограничения: а) должно гореть не менее двух секций; б) не может гореть только две секции в горизонтальном ряду. Нарисуйте *запрещенные* комбинации семафора. Составьте таблицу истинности работы семафора. Запишите логическую функцию, описывающую *запрещенные* комбинации семафора, в дизъюнктивной нормальной форме. Упростите полученное выражение, оставаясь в базисе И, ИЛИ, НЕ. (12 баллов)



Решение:

Запрещенные комбинации				Разрешенные комбинации			
1				9			
2				10			
3				11			
4				12			
5				13			
6				14			
7				15			
8				16			

Обозначим буквами секции семафора: a, b, c – слева направо верхний ряд, d – секция в нижнем ряду. Составим таблицу истинности функции F .

N	a	b	c	d	F
0	0	0	0	0	1
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	0
4	0	1	0	0	1
5	0	1	0	1	0
6	0	1	1	0	1
7	0	1	1	1	0
8	1	0	0	0	1
9	1	0	0	1	0
10	1	0	1	0	1
11	1	0	1	1	0
12	1	1	0	0	1
13	1	1	0	1	0
14	1	1	1	0	0
15	1	1	1	1	0

Функция F истинна (равна 1) на номерах наборов 0,1,2,4,6,8,10 и 12.

$$F = (\bar{a}\bar{b}\bar{c}l\bar{d}) \vee (\bar{a}\bar{b}l\bar{c}ld) \vee (\bar{a}\bar{b}lcl\bar{d}) \vee (\bar{a}\bar{b}l\bar{c}l\bar{d}) \vee (\bar{a}bl\bar{c}l\bar{d}) \vee (\bar{a}blcl\bar{d}) \vee (alb\bar{c}l\bar{d}) \vee (albcl\bar{d})$$

Упростим функцию. По ТИ видно, что $\bar{c}l\bar{d}$ всегда дают $F=1$. Запишем новое выражение:

$$F = (\bar{c}l\bar{d}) \vee (\bar{a}\bar{b}\bar{c}ld) \vee (\bar{a}\bar{b}lcl\bar{d}) \vee (\bar{a}\bar{b}lcl\bar{d}) \vee (alb\bar{c}l\bar{d})$$



<i>N</i>	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>F</i>
0	0	0	0	0	1
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	0
4	0	1	0	0	1
5	0	1	0	1	0
6	0	1	1	0	1
7	0	1	1	1	0
8	1	0	0	0	1
9	1	0	0	1	0
10	1	0	1	0	1
11	1	0	1	1	0
12	1	1	0	0	1
13	1	1	0	1	0
14	1	1	1	0	0
15	1	1	1	1	0

В полученном выражении можно минимизировать выражения в 3 и 4 скобках:

$$(\bar{a}\bar{b}\bar{c}d)\vee(\bar{a}b\bar{c}d)=(\bar{a}c\bar{d}).$$

Итоговое выражение в базисе И, ИЛИ, НЕ: $F = (\bar{c}d)\vee(\bar{a}\bar{b}\bar{c}d)\vee(\bar{a}c\bar{d})\vee(a\bar{b}\bar{c}d)$

4. Иван устроился работать системным администратором в филиал НИИ «Технологии». В первый же день работы Иван получил ответственное задание. Отдел информационных технологий НИИ «Технологии» использует диапазон IP-адресов 192.168.164.0/22, в котором настроено адресное пространство сетей филиалов из 6 подсетей. В результате анализа настроек были установлены адреса компьютеров сетевых администраторов каждой подсети. Известно, что адреса этих компьютеров являются первыми возможными адресами узлов для каждой подсети. Адресное пространство при этом использовано максимально эффективно.

У Ивана имеются следующие познания:

- одному устройству должен соответствовать один IP-адрес вида XXX.XXX.XXX.XXX, при этом XXX – число в диапазоне [0:255], называемое октетом;
- устройства внутри одной подсети должны беспрепятственно взаимодействовать друг с другом и должны группироваться по их расположению;
- выделенный размер подсети содержит минимально необходимое количество адресов;
- подсети большего размера должны находиться в начале адресного диапазона сети;
- в каждом сегменте сети первый адрес зарезервирован под идентификатор подсети, а последний – под широковещательные (broadcast) запросы;
- выделенный размер подсети записывается как 2^n-2 , где n – минимальное количество бит, которое может содержать все номера адресов подсети.

Название подсети	Адрес ПК сетевого администратора	Выделенный размер	Адрес сети	Маска с префиксом	Десятичная маска	Диапазон доступных адресов	Широковещание
А	192.168.164.1						
В	192.168.166.1						
С	192.168.167.1						
Администрация	192.168.167.129						
Пост охраны	192.168.167.161						
IT-отдел	192.168.167.177						



Заполнение информации вызвало у системного администратора затруднения. Необходимо помочь Ивану заполнить таблицу недостающими данными. (20 баллов)

Решение:

Название подсети	Адрес ПК сетевого администратора	Выделенный размер	Адрес сети	Маска с префиксом	Десятичная маска	Диапазон доступных адресов	Широковещание
A	192.168.164.1	510	192.168.164.0	/23	255.255.254.0	192.168.164.1 - 192.168.165.254	192.168.165.255
B	192.168.166.1	254	192.168.166.0	/24	255.255.255.0	192.168.166.1 - 192.168.166.254	192.168.166.255
C	192.168.167.1	126	192.168.167.0	/25	255.255.255.128	192.168.167.1 - 192.168.167.126	192.168.167.127
D	192.168.167.129	30	192.168.167.128	/27	255.255.255.224	192.168.167.129 - 192.168.167.158	192.168.167.159
E	192.168.167.161	14	192.168.167.160	/28	255.255.255.240	192.168.167.161 - 192.168.167.174	192.168.167.175
F	192.168.167.177	2	192.168.167.176	/30	255.255.255.252	192.168.167.177 - 192.168.167.178	192.168.167.179

5. Директор РОНО попросил администратора базы данных принести распечатку со списком детей, одновременно посещающих и секцию шахмат, и секцию настольного тенниса в районном Доме культуры. Администратор базы данных сохранил два упорядоченных списка детей, посещающих каждую секцию. В каждом списке участник представлен одним и тем же уникальным кодом, состоящим из прописных букв латинского алфавита и цифр от 0 до 9. Внутри списка повторений и совпадений нет. Списки упорядочены по возрастанию. Помогите администратору составить алгоритм и программу для решения этой задачи, считая, что списки пригодны для непосредственной обработки выбранным языком программирования и уже загружены в память. Правильность решения докажете на простом примере. В решении **нельзя** использовать встроенные в язык операции, методы или свойства для работы со структурами данных, кроме как для определения длины структуры или добавления (удаления) элементов структуры. Код необходимо снабдить комментариями. (25 баллов)

Решение:

Алгоритм решения задачи:

Задача на пересечение множеств. Пересечение двух упорядоченных множеств **A** и **B** – это множество только с теми элементами **A** и **B**, которые одновременно принадлежат обоим множествам, без дублей. Сложность алгоритма $O(m+n)$, где **m** и **n** – длины входных множеств **A** и **B** соответственно.

Необходимо написать программу, аргументы которой – два упорядоченных массива для пересечения. Функция возвращает третий массив, содержащий элементы, входящие одновременно в оба массива. Необходимо взять два исходных массива и пустой третий. Сделать и обнулить три счетчика индексов массивов: $i=0$, $j=0$, $k=0$. Сравнить верхние (нулевые) элементы массивов. Если $M_1[i] == M_2[j]$, то: $M_3[k] == M_1[i]$, $i=i+1$, $j=j+1$, $k=k+1$. Если $M_1[i] < M_2[j]$, то: $i=i+1$ – поднимаем M_1 вверх на один элемент. Если (иначе, т.к. вариантов больше нет) $M_1[i] > M_2[j]$, то: $j=j+1$ – поднимаем M_2 вверх на один элемент. И так в цикле while пока

$i < L_1$ и $j < L_2$, где L – длина массива. Проверить индексацию массивов и правильность определения верхней границы массива в условии цикла `while`.

Пример решения на PascalABC.NET

```
1 program Zadacha_5; //Необязательный заголовок
2 //Задача на пересечение множеств
3 //В языке есть встроенные множества "set"
4 //и метод a.intersect(b) пересечения множеств
5 //По условиям задачи так делать нельзя и метод
6 //выдает последовательность, а не массив.
7
8 //Динамические массивы с данными для сортировки
9 var
10 a: array of string := ('aa2', 'bb', 'cc', 'ddl', 'ee');
11 b: array of string := ('aa2', 'ddl', 'de', 'ed', 'ff');
12 //Выходной массив. изначально пустой
13 c: array of string;
14 i, j, k: integer; //Счётчики=0 по умолчанию
15
16 begin
17 while (i < a.Length) and (j < b.Length) do
18   if a[i] = b[j] then
19     begin //Ветка true
20       SetLength(c, k + 1);
21       c[k] := a[i];
22       i := i + 1; j := j + 1; k := k + 1;
23     end
24   else //Ветка false
25     if a[i] < b[j] then i := i + 1
26     else j := j + 1;
27   writeln(c);
28
29 //А теперь с помощью встроенного метода
30 a.Intersect(b).Println;
31 end.
```

Окно вывода

```
[aa2, ddl]
aa2 ddl
```

Ответ

6. Василий Иванович решил обложить печь на даче изразцами. Тот бок печи, который выходит в коридор. Жена Василия Ивановича согласилась, но при условии, что изразцы будут квадратными и максимально большого размера. И обязательно чтобы целые, никаких половинок и кусочков для заполнения оставшегося места! Василий Иванович измерил бок печи рулеткой и записал размер: 2640x720 мм. Какого размера изразцы должен заказать Василий Иванович, чтобы выполнить условие супруги? Напишите алгоритм и универсальную программу, которая на основе размеров прямоугольного бока печки вычисляет и выводит в качестве ответа размер стороны изразца, который необходимо заказать. Правильность решения докажете поэтапно вручную по коду. Код необходимо снабдить комментариями. (25 баллов)

Решение:

Алгоритм решения задачи:

Задача на использование рекурсии и алгоритма «Разделяй и властвуй». Есть некоторая поверхность прямоугольной формы с заданными сторонами. Ее необходимо разбить на минимальное количество одинаковых квадратов. Соответственно площадь квадратов должна быть максимальной. Решается с помощью «Алгоритма Евклида». Суть алгоритма: определить базовый случай и применить рекурсию. Базовый случай: соотношение сторон участка 2:1.

Графическая иллюстрация – см. вариант 1. Ответ: сторона изразца = 240 мм. Проверка поэтапно на рисунке с кодом. Проверка по сторонам: $2640:240=11$; $720:240=3$.

Пример решения на PascalABC.NET

```
1 program Zadacha_6; //Необязательный заголовок
2 //Задача на определение максимального квадрата,
3 //которым можно заполнить прямоугольник.
4 var //Переменные для хранения сторон прямоугольника
5 //a,b - стороны участка, m - остаток от деления mod
6 m: integer;
7 //рекурсивная функция
8 function kvadrat(a,b: integer):integer;
9 var //аккумулятор для переменных местами a и b
10 akkum: integer;
11 begin //если указали стороны в неправильном порядке
12 //if a<b then
13 // begin//меняем местами a и b
14 //   akkum:=a;
15 //   a:=b;
16 //   b:=akkum;
17 // end;
18 println('a=',a); //для контроля хода решения
19 println('b=',b);
20 //базовый случай, когда отношение сторон 2/1
21 if (a mod b)=0 then //это и есть базовый случай
22   begin
23     //kvadrat:=b; //можно и так и как ниже
24     Result:=b; //Определяем что возвращает функция
25     Println('Сторона квадрата = ', Result);
26     exit
27   end
28 else //базовый случай не достигнут
29   begin
30     m:=a mod b;
31     //a:=b;
32     //b:=m;
33     kvadrat(b,m); //запускаем рекурсию
34   end;
35 end;
36
37 begin //основная программа
38   kvadrat(2640,720); //вызов рекурсивной функции
39 end.
```

Окно вывода

```
a= 2640
b= 720
a= 720
b= 480
a= 480
b= 240
Сторона квадрата = 240
```



ИНФОРМАТИКА
БИЛЕТ 1. ВАРИАНТ 3

1. С тех пор как в жизни Васи появились электронные таблицы, он непрерывно в них что-то считает. Вот и на каникулах он составил таблицу, показанную на рисунке ниже.

	Q	R	S	T	U	V
17		2				
18	2					
19						
20						
21						
22						

В ячейке S17 записана формула $=\$Q18+R17$ и распространена по строке вправо. В ячейке Q19 записана формула $=Q18+СЧЁТЗ(ДВССЫЛ("Q\$18:\$Q18"))$ и распространена по столбцу вниз. В ячейке R18 записана формула $=СТЕПЕНЬ(\$Q18;2)+R\$17^{\text{LOG}(100)}$ и с ее помощью заполнена вся таблица. Какое значение получит Вася в ячейке V22, если в ней записана формула $\{=\text{МУМНОЖ}(R17:U17;Q18:Q21)-\text{СУММЕСЛИ}(Q18:Q21;">4";R18:U21)\}$? Дайте пояснения. Укажите значения произведения матриц и суммы элементов. **(8 баллов)**

Ответ: 51

Решение:

	2	4	6	8	
2	8	20	40	68	
3	13	25	45	73	
4	20	32	52	80	
5	29	41	61	89	
					51

Произведение матриц = 80. Сумма элементов равна 29.

2. Два гнома решили сыграть в игру «20 драгоценных камней». Они выложили в ряд **20** драгоценных камней. По условию игры можно поочередно брать 1, 2 или 3 камня за раз. Можно брать **любые** камни. Пропускать свой ход нельзя. Выиграет тот, кто заберет последние камни или камень. Напишите алгоритм **выигрышной стратегии** для гнома, ходящего **вторым**. Поясните алгоритм графически. Напишите обобщенный алгоритм для любого четного количества камней. Укажите нижнюю границу применимости алгоритма. **(10 баллов)**

Решение:

Используется Стратегия_2 (любые предметы). Всегда выигрывает Игрок_2 (игрок, делающий ход вторым). **Суть Стратегия_2:**

Ход_1: Игрок_1 берет любое разрешенное количество предметов (1,2 или 3) из любого места ряда.

Ход_2: Игрок_2 (игрок, делающий ход вторым) берет такое количество предметов, чтобы дополнить до 4-х количество предметов, взятых Игрок_1 в предыдущем ходе. **Сумма** предметов **всегда** $\text{Ход}_1+\text{Ход}_2=4$.

И так далее до того момента, когда перед ходом Игрок_2 останется $N<3$ предмета = **Победа** Игрок_2. Проиграть невозможно.

Иллюстрация на рисунке ниже.



Алгоритм подходит для **любого кратного 4-м количества** предметов ($N\%4=0$).
Алгоритм для Игрок_2 (обобщенный): Если $\text{Остаток_предметов} < 3$ – взять все предметы=ПОБЕДА. Иначе взять $(4 - \text{Игрок_1}_{\text{Предыд_ход}})$ предметов.

Нижняя граница – $2 \cdot 4 = 8$ предметов.

	Игрок_1 - ходит первым	Можно брать 1, 2 или 3 ЛЮБЫХ предмета. Всего N предметов, N кратно 4. $N\%4=0$.																							
Вариант 3 Стратегия 2	Игрок_2 - ходит вторым																								
Победитель Игрок 2																									
Исходное состояние		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20				
Ход 1 - любой набор из 1,2 или 3-х предметов		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20				
		3																							
Ход 2 - дополнение хода 1 до 4-х предметов		1	2			6	7	8	9	10	11	12	13	14	15	16	17	18	19	20					
		3+1=4																							
Ход 3 - любой набор из 1,2 или 3-х предметов		1	2				7	8	9	10	11	12	13	14	15	16	17	18	19	20					
		2																							
Ход 4 - дополнение хода 1 до 4-х предметов							7	8	9	10	11	12	13	14	15	16	17	18	19	20					
		2		2																					
Ход 5 - любой набор из 1,2 или 3-х предметов									9	10	11	12	13	14	15	16	17	18	19	20					
		3																							
Ход 6 - дополнение хода 1 до 4-х предметов									9	10	11	12	13	14	15				19	20					
		1											3												
Ход 7 - любой набор из 1,2 или 3-х предметов									9	10	11		13	14	15				19	20					
		1								1															
Ход 8 - дополнение хода 1 до 4-х предметов									10	11		13	14	15					19						
		1								1		1										1			
Ход 9 - любой набор из 1,2 или 3-х предметов										11		13	14	15											
		Независимо от хода Игрок_1 на поле останется $N < 3$ предметов. А значит Игрок_2 сможет забрать их ВСЕ за один ход и выиграть.																							

3. Светофор на перекрестке имеет четыре секции: красную, желтую, зеленую (расположены вертикально, сверху вниз в порядке упоминания), зеленую дополнительную (расположена справа от зеленой основной). Возможны следующие комбинации работающих секций: только красная, только желтая, только зеленая, одновременно красная и желтая, только зеленая дополнительная, одновременно зеленая и зеленая дополнительная, одновременно красная и зеленая дополнительная. Нарисуйте разрешенные комбинации светофора. Составьте таблицу истинности работы светофора. Запишите логическую функцию, описывающую разрешенные комбинации светофора, в дизъюнктивной нормальной форме. Упростите полученное выражение, оставаясь в базисе И, ИЛИ, НЕ. (12 баллов)

**Решение:**

Запрещенные комбинации				Разрешенные комбинации			
0		5		1		2	
6		7		4		3	
10		11		8		12	
13		14		9			
		15					

Обозначим буквами секции светофора: a, b, c – сверху вниз вертикальный ряд от красного до зеленого, d – дополнительная зеленая секция. Составим таблицу истинности функции F .

N	a	b	c	d	F
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	1
4	0	1	0	0	1
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	0
8	1	0	0	0	1
9	1	0	0	1	1
10	1	0	1	0	0
11	1	0	1	1	0
12	1	1	0	0	1
13	1	1	0	1	0
14	1	1	1	0	0
15	1	1	1	1	0

Функция F истинна (равна 1) на номерах наборов 1,2,3,4,8,9 и 12.

$$F = (\bar{a}\bar{b}\bar{c}\bar{d}) \vee (\bar{a}\bar{b}c\bar{d}) \vee (\bar{a}\bar{b}c\bar{d}) \vee (\bar{a}\bar{b}c\bar{d}) \vee (a\bar{b}\bar{c}\bar{d}) \vee (a\bar{b}\bar{c}d) \vee (a\bar{b}c\bar{d})$$

Упростим функцию. По ТИ видно, что в наборах 2-3 и 8-9 можно попарно сократить переменную d , а в наборах 4-12 сократить переменную a . Запишем новое выражение:

$$F = (\bar{a}\bar{b}\bar{c}\bar{d}) \vee (\bar{a}\bar{b}c) \vee (b\bar{c}\bar{d}) \vee (a\bar{b}c)$$

Итоговое выражение в базисе И, ИЛИ, НЕ.



<i>N</i>	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>F</i>
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	1
4	0	1	0	0	1
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	0
8	1	0	0	0	1
9	1	0	0	1	1
10	1	0	1	0	0
11	1	0	1	1	0
12	1	1	0	0	1
13	1	1	0	1	0
14	1	1	1	0	0
15	1	1	1	1	0

4. Иван устроился работать системным администратором в филиал НИИ «Технологии». В первый же день работы Иван получил ответственное задание. Отдел информационных технологий НИИ «Технологии» использует диапазон IP-адресов 192.168.168.0/22, в котором настроено адресное пространство сетей филиалов из 6 подсетей. В результате анализа настроек были установлены адреса компьютеров сетевых администраторов каждой подсети. Известно, что адреса этих компьютеров являются первыми возможными адресами узлов для каждой подсети. Адресное пространство при этом использовано максимально эффективно.

У Ивана имеются следующие познания:

- одному устройству должен соответствовать один IP-адрес вида XXX.XXX.XXX.XXX, при этом XXX – число в диапазоне [0:255], называемое октетом;
- устройства внутри одной подсети должны беспрепятственно взаимодействовать друг с другом и должны группироваться по их расположению;
- выделенный размер подсети содержит минимально необходимое количество адресов;
- подсети большего размера должны находиться в начале адресного диапазона сети;
- в каждом сегменте сети первый адрес зарезервирован под идентификатор подсети, а последний – под широковещательные (broadcast) запросы;
- выделенный размер подсети записывается как $2^n - 2$, где n – минимальное количество бит, которое может содержать все номера адресов подсети.



Название подсети	Адрес ПК сетевого администратора	Выделенный размер	Адрес сети	Маска с префиксом	Десятичная маска	Диапазон доступных адресов	Широковещание
А	192.168.168.1						
В	192.168.170.1						
С	192.168.171.1						
Администрация	192.168.171.129						
Пост охраны	192.168.171.161						
IT-отдел	192.168.171.177						

Заполнение информации вызвало у системного администратора затруднения. Необходимо помочь Ивану заполнить таблицу недостающими данными. **(20 баллов)**

Решение:

Название подсети	Адрес ПК сетевого администратора	Выделенный размер	Адрес сети	Маска с префиксом	Десятичная маска	Диапазон доступных адресов	Широковещание
А	192.168.168.1	510	192.168.168.0	/23	255.255.254.0	192.168.168.1 - 192.168.169.254	192.168.169.255
В	192.168.170.1	254	192.168.170.0	/24	255.255.255.0	192.168.170.1 - 192.168.166.254	192.168.170.255
С	192.168.171.1	126	192.168.171.0	/25	255.255.255.128	192.168.171.1 - 192.168.171.126	192.168.171.127
Д	192.168.171.129	30	192.168.171.128	/27	255.255.255.224	192.168.171.129 - 192.168.171.158	192.168.171.159
Е	192.168.171.161	14	192.168.171.160	/28	255.255.255.240	192.168.171.161 - 192.168.171.174	192.168.171.175
Ф	192.168.171.177	2	192.168.171.176	/30	255.255.255.252	192.168.171.177 - 192.168.171.178	192.168.171.179

5. Директор самой большой школы города на 2000 учеников попросил учителя информатики выяснить список учеников школы, которые приняли участие в городской олимпиаде по информатике. Директор дал учителю на флешке два файла: в одном список учеников школы, в другом список участников олимпиады. Оба списка представляют собой упорядоченный набор уникальных кодов учеников. Каждый ученик закодирован в списке заглавными буквами русского алфавита и цифрой от 1 до 8. Помогите учителю информатики составить алгоритм и программу для решения этой задачи, считая, что списки пригодны для непосредственной обработки выбранным языком программирования и уже загружены в память. Правильность решения докажете на простом примере. В решении **нельзя** использовать встроенные в язык операции, методы или свойства для работы со структурами данных, кроме как для определения длины структуры или добавления (удаления) элементов структуры. Код необходимо снабдить комментариями. **(25 баллов)**

Решение:**Алгоритм решения задачи:**

Задача на пересечение множеств. Пересечение двух упорядоченных множеств **A** и **B** – это множество только с теми элементами **A** и **B**, которые одновременно принадлежат обоим множествам, без дублей. Сложность алгоритма $O(m+n)$, где **m** и **n** — длины входных множеств **A** и **B** соответственно.

Необходимо написать программу, аргументы которой – два упорядоченных массива для пересечения. Функция возвращает третий массив, содержащий элементы, входящие одновременно в оба массива. Необходимо взять два исходных массива и пустой третий. Сделать и обнулить три счетчика индексов массивов: $i=0, j=0, k=0$. Сравнить верхние (нулевые) элементы массивов. Если $M_{1i}=M_{2j}$, то: $M_{3k}=M_{1i}, i=i+1, j=j+1, k=k+1$. Если $M_{1i}<M_{2j}$, то: $i=i+1$ – поднимаем M_1 вверх на один элемент. Если (иначе, т.к. вариантов больше нет) $M_{1i}>M_{2j}$, то: $j=j+1$ – поднимаем M_2 вверх на один элемент. И так в цикле **while** пока $i<L_1$ и $j<L_2$, где L – длина массива. Проверить индексацию массивов и правильность определения верхней границы массива в условии цикла **while**.

Пример решения на PascalABC.NET:

```
1 program Zadacha_5; //Необязательный заголовок
2 //Задача на пересечение множеств
3 //В языке есть встроенные множества "set"
4 //и метод a.intersect(b) пересечения множеств
5 //По условиям задачи так делать нельзя и метод
6 //выдает последовательность, а не массив.
7
8 //Динамические массивы с данными для сортировки
9 var
10 a: array of string := ('AA2', 'BB', 'BB', 'ГГ1', 'ДД');
11 b: array of string := ('AA2', 'ГГ1', 'ДЕ', 'ЕД', 'ЖЖ');
12 //Выходной массив. изначально пустой
13 c: array of string;
14 i, j, k: integer; //Счётчики=0 по умолчанию
15
16 begin
17 while (i < a.Length) and (j < b.Length) do
18   if a[i] = b[j] then
19     begin //Ветка true
20       SetLength(c, k + 1);
21       c[k] := a[i];
22       i := i + 1; j := j + 1; k := k + 1;
23     end
24   else //Ветка false
25     if a[i] < b[j] then i := i + 1
26     else j := j + 1;
27   writeln(c);
28
29 //А теперь с помощью встроенного метода
30 a.Intersect(b).Println;
31 end.
```

Окно вывода

```
[AA2, ГГ1]
AA2 ГГ1
```

Ответ



6. Бабушка испекла на день рождения Машеньки пирог с яблоками. Форма для пирога у бабушки прямоугольная, размером 40x30 см. Машенька разделила пирог на одинаковые квадраты и угостила всех гостей, не забыв про себя. От пирога ничего не осталось, даже крошки. Всем достались одинаковые куски. Сколько человек было на дне рождения у Машеньки, включая ее саму? Напишите алгоритм и универсальную программу, которая на основе размеров формы для пирога вычисляет и выводит в качестве ответа размер стороны пирога и количество гостей, включая Машеньку. Правильность решения докажите поэтапно вручную по коду. Правильность решения докажите поэтапно вручную по коду. Код необходимо снабдить комментариями. **(25 баллов)**

Решение:

Алгоритм решения задачи

Задача на использование рекурсии и алгоритма «Разделяй и властвуй». Есть некоторая поверхность прямоугольной формы с заданными сторонами. Ее необходимо разбить на минимальное количество одинаковых квадратов. Соответственно площадь квадратов должна быть максимальной. Решается с помощью «Алгоритма Евклида». Суть алгоритма: определить базовый случай и применить рекурсию. Базовый случай: соотношение сторон участка 2:1. Графическая иллюстрация см. вариант 1. Ответ: сторона куска = 10 см. Проверка поэтапно на рисунке с кодом. Проверка по сторонам: $40:10=4$; $30:10=3$; $4 \times 3=12$ гостей, включая Машеньку.

Пример решения на PascalABC.NET

```

1 program Zadacha_6; //Необязательный заголовок
2 //Задача на определение максимального квадрата,
3 //которым можно заполнить прямоугольник.
4 var //Переменные для хранения сторон прямоугольника
5 //a,b - стороны участка, m - остаток от деления mod
6   a,b,l,L: integer;
7 //рекурсивная функция
8 function kvadrat(a,b: integer):integer;
9 var m: integer;
10 begin //если указали стороны в неправильном порядке
11   println('a=',a); //для контроля хода решения
12   println('b=',b);
13   //базовый случай, отношение сторон 2/1 для
14   //подобранных примеров
15   if (a mod b)=0 then //это и есть базовый случай
16     begin
17       Result:=b; //возвращаемое значение
18       Println('Сторона квадрата = ', b);
19     end
20   else //базовый случай не достигнут
21     begin
22       m:=a mod b;
23       Result:=kvadrat(b,m); //запускаем рекурсию
24     end;
25   end;
26
27 begin //основная программа
28   a:=40; b:=30;
29   L:=kvadrat(a,b); //вызов рекурсивной функции
30   Println('Ответ = ', (a/L)*(b/L));
31 end.
```

Окно вывода

```
a= 40
b= 30
a= 30
b= 10
Сторона квадрата = 10
Ответ = 12
```



ИНФОРМАТИКА
БИЛЕТ 1. ВАРИАНТ 4

1. С тех пор как в жизни Васи появились электронные таблицы, он непрерывно в них что-то считает. Вот и на каникулах он составил таблицу, показанную на рисунке ниже.

	Q	R	S	T	U	V
17		-2				
18	-2					
19						
20						
21						
22						

В ячейке S17 записана формула $=\$Q18+R17$ и распространена по строке вправо. В ячейке Q19 записана формула $=Q18+СЧЁТЗ(ДВССЫЛ("Q\$18:\$Q18"))$ и распространена по столбцу вниз. В ячейке R18 записана формула $=СТЕПЕНЬ(\$Q18;2)+R\$17^{\text{LOG}(100)}$ и с ее помощью заполнена вся таблица. Какое значение получит Вася в ячейке V22, если в ней записана формула $\{=МУМНОЖ(R17:U17;Q18:Q21)-СУММЕСЛИ(Q18:Q21;">0";R18:U21)\}$? Дайте пояснения. Укажите значения произведения матриц и суммы элементов. **(8 баллов)**

Ответ: -5

Решение:

	-2	-4	-6	-8	
-2	8	20	40	68	
-1	5	17	37	65	
0	4	16	36	64	
1	5	17	37	65	
					-5

Произведение матриц = 0. Сумма элементов равна 5.

2. Два гнома решили сыграть в игру «21 драгоценный камень». Они выложили в ряд **21** драгоценный камень. По условию игры можно поочередно брать 1, 2 или 3 камня за раз. Можно брать **любые** камни. Пропускать свой ход нельзя. Выиграет тот, кто заберет последние камни или камень. Напишите алгоритм **выигрышной стратегии** для гнома, ходящего **первым**. Поясните алгоритм графически. Напишите обобщенный алгоритм для любого нечетного количества камней. Укажите нижнюю границу применимости алгоритма. **(10 баллов)**

Решение:

Используется Стратегия_2 (любые предметы). Всегда выигрывает Игрок_1 (игрок, делающий ход первым). Используется Стратегия_2 (любые предметы). Всегда выигрывает Игрок_2 (игрок, делающий ход вторым). **Суть Стратегия_2:**

Ход_1: Игрок_1 берет любое разрешенное количество предметов (1,2 или 3) из любого места ряда таким образом, чтобы $\text{Остаток}_{\text{ход}_1} \% 4 = 0$ стал кратным 4.

Ход_2: получили вариант_3, только игроки как бы поменялись очередностью ходов. Поэтому используется Стратегия_2 для Игрок_1. Проиграть невозможно. Иллюстрация на рисунке ниже.

Алгоритм подходит для **любого $m \cdot 4 + (1-3)$ количества** предметов ($N \% 4 \neq 0$, $N \% 4 = 1$, или $=2$, или $=3$).



Алгоритм для Игрок_1(обобщённый): Если Ход_1: 1, 2 или 3 предмета так, чтобы $(N-(1-3))\%4=0$. Если Остаток_предметов <3 – взять все предметы=ПОБЕДА. Иначе взять (4-Игрок_2_{Предыд_ход}) предметов.

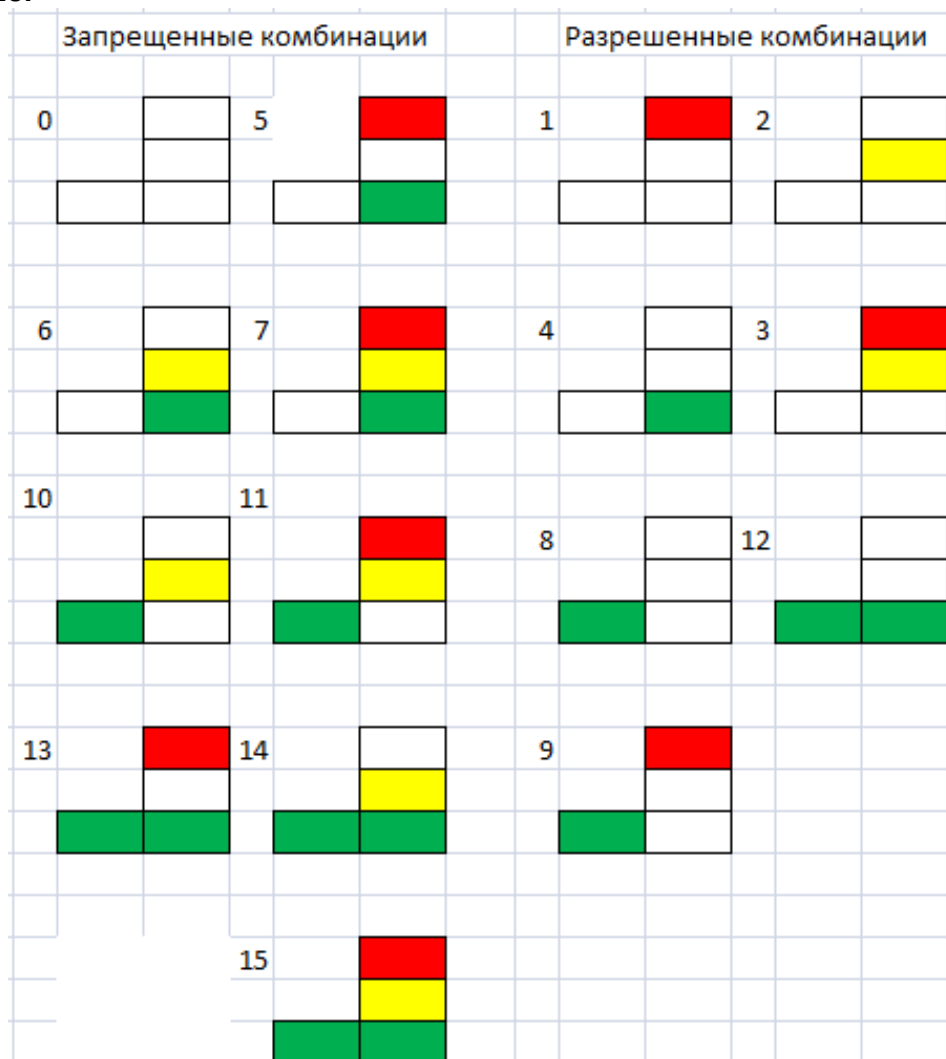
Нижняя граница – $2*4+1=9$ предметов.

	Игрок_1 - ходит первым										<div style="border: 1px solid black; padding: 5px;"> Можно брать 1, 2 или 3 ЛЮБЫХ предмета. Всего N предметов, N !кратно 4. $N\%4\neq 0$ (м.б. 1,2 или 3). </div>												
Вариант 4 Стратегия 2	Игрок_2 - ходит вторым																						
Победитель Игрок 1																							
Исходное состояние	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21		
Ход 1 - любой набор из 1,2 или 3-х предметов, чтобы оставить $N\%4=0$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21		
											1	$21\%4=1$											
Ход 2 - любой набор из 1,2 или 3-х предметов	1	2	3	4	5	6	7	8	9	10		12	13	14	15	16	17	18	19	20	21		
				1											1				1				
Ход 3 - дополнение хода 2 до 4-х предметов	1	2	3		5	6	7	8	9	10		12	13	14		16	17	18		20	21		
				1				1											1				1
Далее игра сводится к Стратегия_2 из Вариант_3 до победы Игрок_1																							

3. Светофор на перекрестке имеет четыре секции: красную, желтую, зеленую (расположены вертикально, сверху вниз в порядке упоминания), зеленую дополнительную (расположена справа от зеленой основной). Возможны следующие комбинации работающих секций: только красная, только желтая, только зеленая, одновременно красная и желтая, только зеленая дополнительная, одновременно зеленая и зеленая дополнительная, одновременно красная и зеленая дополнительная. Нарисуйте *запрещенные* комбинации светофора. Составьте таблицу истинности работы светофора. Запишите логическую функцию, описывающую *запрещенные* комбинации светофора, в дизъюнктивной нормальной форме. Упростите полученное выражение, оставаясь в базисе И, ИЛИ, НЕ. **(12 баллов)**



Решение:



Обозначим буквами секции светофора: a, b, c – сверху вниз вертикальный ряд от красного до зеленого, d – дополнительная зеленая секция. Составим таблицу истинности функции F .

N	a	b	c	d	F
0	0	0	0	0	1
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	0
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	1
7	0	1	1	1	1
8	1	0	0	0	0
9	1	0	0	1	0
10	1	0	1	0	1
11	1	0	1	1	1
12	1	1	0	0	0
13	1	1	0	1	1
14	1	1	1	0	1
15	1	1	1	1	1



Функция F истинна (равна 1) на номерах наборов 0,5,6,7,10,11,13,14 и 15.

$F = (\bar{a}\bar{b}\bar{c}\bar{d}) \vee (\bar{a}b\bar{c}d) \vee (\bar{a}b\bar{c}\bar{d}) \vee (\bar{a}b\bar{c}d) \vee (a\bar{b}\bar{c}\bar{d}) \vee (a\bar{b}\bar{c}d) \vee (a\bar{b}\bar{c}d) \vee (a\bar{b}\bar{c}\bar{d}) \vee (a\bar{b}c\bar{d}) \vee (a\bar{b}cd) \vee (a\bar{b}c\bar{d}) \vee (a\bar{b}cd)$
 Упростим функцию. По ТИ видно, что в наборах 6-7, 10-11 и 14-15 можно попарно сократить переменную d , а в наборах 5-13 сократить переменную a . Запишем новое выражение: $F = (\bar{a}\bar{b}\bar{c}\bar{d}) \vee (b\bar{c}\bar{d}) \vee (\bar{a}b\bar{c}) \vee (a\bar{b}\bar{c}) \vee (a\bar{b}c)$.

N	a	b	c	d	F
0	0	0	0	0	1
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	0
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	1
7	0	1	1	1	1
8	1	0	0	0	0
9	1	0	0	1	0
10	1	0	1	0	1
11	1	0	1	1	1
12	1	1	0	0	0
13	1	1	0	1	1
14	1	1	1	0	1
15	1	1	1	1	1

В полученном выражении можно минимизировать выражения в 4 и 5 скобках:

$$(a\bar{b}\bar{c}) \vee (a\bar{b}c) = (a\bar{c}).$$

Итоговое выражение в базисе И, ИЛИ, НЕ: $F = (\bar{a}\bar{b}\bar{c}\bar{d}) \vee (b\bar{c}\bar{d}) \vee (\bar{a}b\bar{c}) \vee (a\bar{c}).$

4. Иван устроился работать системным администратором в филиал НИИ «Технологии». В первый же день работы Иван получил ответственное задание. Отдел информационных технологий НИИ «Технологии» использует диапазон IP-адресов 192.168.172.0/22, в котором настроено адресное пространство сетей филиалов из 6 подсетей. В результате анализа настроек были установлены адреса компьютеров сетевых администраторов каждой подсети. Известно, что адреса этих компьютеров являются первыми возможными адресами узлов для каждой подсети. Адресное пространство при этом использовано максимально эффективно.

У Ивана имеются следующие познания:

- одному устройству должен соответствовать один IP-адрес вида XXX.XXX.XXX.XXX, при этом XXX – число в диапазоне [0:255], называемое октетом;
- устройства внутри одной подсети должны беспрепятственно взаимодействовать друг с другом и должны группироваться по их расположению;
- выделенный размер подсети содержит минимально необходимое количество адресов;
- подсети большего размера должны находиться в начале адресного диапазона сети;
- в каждом сегменте сети первый адрес зарезервирован под идентификатор подсети, а последний – под широковещательные (broadcast) запросы;
- выделенный размер подсети записывается как $2^n - 2$, где n – минимальное количество бит, которое может содержать все номера адресов подсети.



Название подсети	Адрес ПК сетевого администратора	Выделенный размер	Адрес сети	Маска с префиксом	Десятичная маска	Диапазон доступных адресов	Широковещание
А	192.168.172.1						
В	192.168.174.1						
С	192.168.175.1						
Администрация	192.168.175.129						
Пост охраны	192.168.175.161						
IT-отдел	192.168.175.177						

Заполнение информации вызвало у системного администратора затруднения. Необходимо помочь Ивану заполнить таблицу недостающими данными. (20 баллов)

Решение:

Название подсети	Адрес ПК сетевого администратора	Выделенный размер	Адрес сети	Маска с префиксом	Десятичная маска	Диапазон доступных адресов	Широковещание
А	192.168.172.1	510	192.168.172.0	/23	255.255.254.0	192.168.172.1 - 192.168.173.254	192.168.173.255
В	192.168.174.1	254	192.168.174.0	/24	255.255.255.0	192.168.174.1 - 192.168.174.254	192.168.174.255
С	192.168.175.1	126	192.168.175.0	/25	255.255.255.128	192.168.175.1 - 192.168.175.126	192.168.175.127
Д	192.168.175.129	30	192.168.175.128	/27	255.255.255.224	192.168.175.129 - 192.168.175.158	192.168.175.159
Е	192.168.175.161	14	192.168.175.160	/28	255.255.255.240	192.168.175.161 - 192.168.175.174	192.168.175.175
F	192.168.175.177	2	192.168.175.176	/30	255.255.255.252	192.168.175.177 - 192.168.175.178	192.168.175.179

5. Отдел закупок шахты «Первая марсианская» получил список доступных запчастей для горно-шахтных погрузчиков, имеющих в наличии на базе снабжения «Лунная 4». В отделе уже был список потребных запчастей от главного инженера шахты. Начальник отдела закупок попросил старшего программиста выяснить, какие из потребных запчастей имеются в наличии на базе снабжения, и прислать ему такой список по квантовой почте в виде упорядоченной последовательности артикулов запчастей. Каждый артикул записан в списках заглавными буквами английского алфавита и цифрой от 1 до 3. Помогите старшему программисту шахты составить алгоритм и программу для решения этой задачи, считая, что исходные списки пригодны для непосредственной обработки выбранным языком программирования и уже загружены в память. Правильность решения докажете на простом примере. В решении **нельзя** использовать встроенные в язык операции, методы или свойства для работы со структурами данных, кроме как для определения длины структуры или добавления (удаления) элементов структуры. Код необходимо снабдить комментариями. (25 баллов)

Решение:

Алгоритм решения задачи:

Задача на пересечение множеств. Пересечение двух упорядоченных множеств **А** и **В** – это множество только с теми элементами **А** и **В**, которые одновременно принадлежат обоим

множествам, без дублей. Сложность алгоритма $O(m+n)$, где m и n — длины входных множеств A и B соответственно.

Необходимо написать программу, аргументы которой – два упорядоченных массива для пересечения. Функция возвращает третий массив, содержащий элементы, входящие одновременно в оба массива. Необходимо взять два исходных массива и пустой третий. Сделать и обнулить три счетчика индексов массивов: $i=0$, $j=0$, $k=0$. Сравнить верхние (нулевые) элементы массивов. Если $M_{1i}=M_{2j}$, то: $M_{3k}=M_{1i}$, $i=i+1$, $j=j+1$, $k=k+1$. Если $M_{1i}<M_{2j}$, то: $i=i+1$ – поднимаем M_1 вверх на один элемент. Если (иначе, т.к. вариантов больше нет) $M_{1i}>M_{2j}$, то: $j=j+1$ – поднимаем M_2 вверх на один элемент. И так в цикле `while` пока $i<L_1$ и $j<L_2$, где L – длина массива. Проверить индексацию массивов и правильность определения верхней границы массива в условии цикла `while`.

Пример решения на PascalABC.NET

```
1 program Zadacha_5; //Необязательный заголовок
2 //Задача на пересечение множеств
3 //В языке есть встроенные множества "set"
4 //и метод a.intesect(b) пересечения множеств
5 //По условиям задачи так делать нельзя и метод
6 //выдает последовательность, а не массив.
7
8 //Динамические массивы с данными для сортировки
9 var
10 a: array of string := ('AA2', 'BB', 'CC', 'DD1', 'EE');
11 b: array of string := ('AA2', 'DD1', 'DE', 'ED', 'FF');
12 //Выходной массив. изначально пустой
13 c: array of string;
14 i, j, k: integer; //Счётчики=0 по умолчанию
15
16 begin
17 while (i < a.Length) and (j < b.Length) do
18   if a[i] = b[j] then
19     begin //Ветка true
20       SetLength(c, k + 1);
21       c[k] := a[i];
22       i := i + 1; j := j + 1; k := k + 1;
23     end
24   else //Ветка false
25     if a[i] < b[j] then i := i + 1
26     else j := j + 1;
27   writeln(c);
28
29 //А теперь с помощью встроенного метода
30 a.Intersect(b).Println;
31 end.
```

Окно вывода

```
[AA2, DD1]
AA2 DD1
```

← Ответ

6. Комитет по благоустройству города N решил засадить прямоугольную лужайку в городском парке культуры и отдыха двумя оттенками травы в шахматном порядке. Чтобы получилось красиво, решили не мельчить и сделать каждый квадрат максимально большим и таким, чтобы пустого места не осталось. Главный дизайнер комитета попросил в отделе информационного сопровождения написать универсальную программу для расчета стороны такого квадрата. Размер лужайки: 34x22 м. Напишите алгоритм и универсальную программу, которая на основе размеров лужайки вычисляет и выводит в качестве ответа размер стороны квадрата для засева травой. Правильность решения докажите поэтапно вручную по коду. Правильность решения докажите поэтапно вручную по коду. Код необходимо снабдить комментариями. (25 баллов)

Решение:

Алгоритм решения задачи

Задача на использование рекурсии и алгоритма «Разделяй и властвуй». Есть некоторая поверхность прямоугольной формы с заданными сторонами. Ее необходимо разбить на минимальное количество одинаковых квадратов. Соответственно площадь квадратов должна быть максимальной. Решается с помощью «Алгоритма Евклида». Суть алгоритма: определить базовый случай и применить рекурсию. Базовый случай: соотношение сторон участка 2:1. Графическая иллюстрация см. вариант 1. Ответ: сторона квадрата = 2 м. Проверка поэтапно на рисунке с кодом. Проверка по сторонам: $34:2=17$; $22:2=11$.

Пример решения на PascalABC.NET

```
1 program Zadacha_6; //Необязательный заголовок
2 //Задача на определение максимального квадрата,
3 //которым можно заполнить прямоугольник.
4 var //Переменные для хранения сторон прямоугольника
5 //a,b - стороны участка, m - остаток от деления mod
6 m: integer;
7 //рекурсивная функция
8 function kvadrat(a,b: integer):integer;
9 var //аккумулятор для перемены местами a и b
10 akkum: integer;
11 begin //если указали стороны в неправильном порядке
12 //if a<b then
13 // begin//меняем местами a и b
14 // akkum:=a;
15 // a:=b;
16 // b:=akkum;
17 // end;
18 println('a=',a); //для контроля хода решения
19 println('b=',b);
20 //базовый случай, когда отношение сторон 2/1
21 if (a mod b)=0 then //это и есть базовый случай
22 begin
23 //kvadrat:=b; //можно и так и как ниже
24 Result:=b; //Определим что возвращает функция
25 Println('Сторона квадрата = ', Result);
26 exit
27 end
28 else //базовый случай не достигнут
29 begin
30 m:=a mod b;
31 //a:=b;
32 //b:=m;
33 kvadrat(b,m); //запускаем рекурсию
34 end;
35 end;
36
37 begin //основная программа
38 kvadrat(34,22); //вызов рекурсивной функции
39 end.
```

Окно вывода

```
a= 34
b= 22
a= 22
b= 12
a= 12
b= 10
a= 10
b= 2
Сторона квадрата = 2
```


**ИНФОРМАТИКА****БИЛЕТ 2. ВАРИАНТ 1**

1. Пропуски в числовых массивах – кошмар перфекциониста Володи. Помогите ему заполнить ячейки, выделенные красным, желтым и зеленым цветами (или соответствующим узором). Определить значения можно по следующим формулам:

Зеленая (узор «вертикальные линии»): =ЕСЛИ(RC9=1;R[8]C[-1]+R[9]C[-3]*2;ЕСЛИ(R[1]C9=2;(R[6]C[-5]/4+R[5]C[-1]/3)+3;(RC[-9]^2-R10C15*3)/6))

Красная (узор «клетка»): =ЕСЛИ(ИЛИ(R7C[-4]>R[-2]C15;И(R6C<=R[3]C14;R4C17>50;-R[-2]C[5]<0));R[4]C9*(1-R[5]C[-4])/R[4]C[-4]+1;R[-4]C[-3]+R[6]C[2]/R4C[1])

Желтая (узор «горизонтальные линии»): =ЕСЛИ(R[5]C[4]<R10C10;R[2]C[7]/R[3]C[7]^2;ЕСЛИ(И(RC[4]-R[7]C/R12C[9]<0;R[2]C12>75);2;1))

Значения цветных ячеек (ячеек с узором) необходимо округлять до ближайшего целого числа. Дайте пояснения.

	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
2	3	93	23	27	2	35	43	54	22	54	61	76	5	79	96
3	15	80	66	7	78	30	93	65	1	10	21	10	14	69	98
4	17	58	5	69	35	60	10	16	95	68	19	64	100	23	58
5	22	12	45	60	56	85	62	29	42	3	57	91	28	12	20
6	65	48	86	68	61	53	82	27	15	48	5	57	4	4	29
7	77	3	82	71	26	37	4	95	8	51	40	17	36		
8	73	37	85	55	13	59	61	18	3	12	57	84	58	74	12
9	77	21	27	33	40	35	48	88	25	17	47	21	75	35	
10	93	52	80	76	13	45	60	80	3	4	25	17	2	69	46
11	99	25	31	88	74	57	72	80	23	96	52	47	88	49	64
12	37	1	40	78	85	87	71	97	12	15	17	39	78	80	95
13	55	97	62	95	77	89	25	45	42	86	54	21	82	13	30
14	71	69	81	100	70	99	13	90	44	14	59	29	72	55	60
15	61	12	38	15	91	3	79	50	34	43	50	67	10	33	15
16	27	31	58	18	9	17	56	73	4	28	14	31	37	80	46

(8 баллов)

Ответ:

Зеленая (узор «вертикальные линии»): -1

Красная (узор «клетка»): -25

Желтая (узор «горизонтальные линии»): 1

Решение:

	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
2	3	93	23	27	2	35	43	54	22	54	61	76	5	79	96
3	15	80	66	7	78	30	93	65	1	10	21	10	14	69	98
4	17	58	5	69	35	60	10	16	95	68	19	64	100	23	58
5	22	12	45	60	56	85	62	29	42	3	57	91	28	12	20
6	65	48	86	68	61	53	82	27	15	48	5	57	4	4	29
7	77	3	82	1	71	26	37	4	95	8	51	40	-1	17	36
8	73	37	85	55	13	59	61	18	3	12	57	84	58	74	12
9	77	21	27	33	40	35	48	-25	88	25	17	47	21	75	35
10	93	52	80	76	13	45	60	80	3	4	25	17	2	69	46
11	99	25	31	88	74	57	72	80	23	96	52	47	88	49	64
12	37	1	40	78	85	87	71	97	12	15	17	39	78	80	95
13	55	97	62	95	77	89	25	45	42	86	54	21	82	13	30
14	71	69	81	100	70	99	13	90	44	14	59	29	72	55	60
15	61	12	38	15	91	3	79	50	34	43	50	67	10	33	15
16	27	31	58	18	9	17	56	73	4	28	14	31	37	80	46



2. Два гнома решили сыграть в игру «20 драгоценных камней». Они выложили в ряд 20 драгоценных камней. По условию игры можно поочередно брать 1, 2 или 3 камня за раз. Можно брать **любые** камни. Пропускать свой ход нельзя. **Проигрывает** тот, кто заберет **последние** камни или камень. Напишите алгоритм **выигрышной стратегии** для гнома, ходящего **первым**. Поясните алгоритм графически. Напишите обобщенный алгоритм для любого четного количества камней. Укажите нижнюю границу применимости алгоритма. **(10 баллов)**

Решение:

Условие выигрыша Игрок_1 - $N\%(k+1) \neq 1$ + Стратегия_1.

Условие выигрыша Игрок_2 - $N\%(k+1) = 1$ + Стратегия_2. (стратегия_2 описана в следующем варианте).

Стратегия: оставить противнику такой Остаток предметов, чтобы $Остаток\%(k+1) = 1$.

$20\%4 = 0 \neq 1$. Поэтому используется Стратегия_1. Всегда выигрывает Игрок_1 (игрок, делающий ход первым). **Суть Стратегия_1:**

Ход_1: Взять столько предметов, чтобы $Остаток\%(k+1) = 1$. Формула Остатка: $1 + i*(k+1)$, где $i = 1, 2, 3, 4, 5, \dots$. Берем 3, остаток 17, $17\%4 = 1$.

Ход_2: Игрок_2 (игрок, делающий ход вторым) берет любое разрешенное количество предметов.

Реализовать Стратегия_1 Игрок_2 не может, так как следующая выигрышная комбинация - 13, а для этого нужно взять 4 предмета, что больше разрешенных 3-х. Игрок_2 обречен. Дальнейшие ходы повторяют Ход_1 и Ход_2.

Алгоритм для Игрок_1 (обобщенный): Шаг 1 - $Остаток\%(k+1) = 1$. Повтор до Победа.

Нижняя граница - 2 предмета.

	Игрок_1 - ходит первым	Можно брать 1, 2 или 3 ЛЮБЫХ предмета. Всего N предметов, $N\%(3+1) \neq 1$.	k=3
Вариант 1 Стратегия 1	Игрок_2 - ходит вторым		
Победитель Игрок 1			5
Исходное состояние	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20		9
Ход 1 - забрал 1,2 и 3	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20	Остаток=17, $17\%(3+1)=1$	13
Ход 2 - любой набор из 1,2 или 3-х предметов	4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20	Остаток=14, $14\%(3+1)=2$	17
Ход 3 - делаем остаток 1	7 8 9 10 11 12 13 14 15 16 17 18 19 20	Остаток=13, $13\%(3+1)=1$	
Ход 4 - любой набор из 1,2 или 3-х предметов	8 9 10 11 12 13 14 15 16 17 18 19 20	Остаток=12, $12\%(3+1)=0$	
Ход 5 - делаем остаток 1	9 10 11 12 13 14 15 16 17 18 19 20	Остаток=9, $9\%(3+1)=1$	
Ход 6 - любой набор из 1,2 или 3-х предметов	12 13 14 15 16 17 18 19 20	Остаток=6, $6\%(3+1)=2$	
Ход 7 - делаем остаток 1	15 16 17 18 19 20	Остаток=5, $5\%(3+1)=1$	
Далее вариантов нет. В любой комбинации Игрок_1 сможет оставить 1 предмет. Это Победа.			



3. На площадке размещения техники в открытом угольном карьере специальный светофор для карьерных самосвалов состоит из 4 секций, расположенных буквой Т (горизонтальная составляющая содержит три секции, а вертикальная две). Каждая секция может мигать оранжевым светом или быть выключенной. На работу светофора наложены следующие ограничения: а) должно гореть не менее двух секций; б) не может гореть только две секции в горизонтальном ряду. Нарисуйте разрешенные комбинации светофора. Составьте таблицу истинности работы светофора. Запишите логическую функцию, описывающую разрешенные комбинации светофора, в дизъюнктивной нормальной форме. Упростите полученное выражение, оставаясь в базисе И, ИЛИ, НЕ. (12 баллов)

Решение:

Запрещенные комбинации			Разрешенные комбинации		
1			9		
2			10		
3			11		
4			12		
5			13		
6			14		
7			15		
8			16		

1. Обозначим буквами секции светофора: a, b, c – слева направо верхний ряд, d – секция в нижнем ряду. Составим таблицу истинности функции F .

N	a	b	c	d	F
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	0
7	0	1	1	1	1
8	1	0	0	0	0
9	1	0	0	1	1
10	1	0	1	0	0
11	1	0	1	1	1
12	1	1	0	0	0
13	1	1	0	1	1
14	1	1	1	0	1
15	1	1	1	1	1



Функция F истинна (равна 1) на номерах наборов 3,5,7,9,11,13,14 и 15.

$$F = (\bar{a}\bar{b}lcld) \vee (\bar{a}bl\bar{c}ld) \vee (\bar{a}blcld) \vee (a\bar{b}l\bar{c}ld) \vee (a\bar{b}lcld) \vee (alb\bar{c}ld) \vee (albcl\bar{d}) \vee (albclsd)$$

Упростим функцию. По ТИ видно, что $clsd$ всегда дают $F=1$. Запишем новое выражение:

$$F = (clsd) \vee (\bar{a}bl\bar{c}ld) \vee (a\bar{b}l\bar{c}ld) \vee (alb\bar{c}ld) \vee (albcl\bar{d})$$

N	a	b	c	d	F
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	0
7	0	1	1	1	1
8	1	0	0	0	0
9	1	0	0	1	1
10	1	0	1	0	0
11	1	0	1	1	1
12	1	1	0	0	0
13	1	1	0	1	1
14	1	1	1	0	1
15	1	1	1	1	1

В полученном выражении можно минимизировать выражения в 3 и 4 скобках: $(a\bar{b}l\bar{c}ld) \vee (alb\bar{c}ld) = (a\bar{c}ld)$. Дальше минимизировать нечего. Итоговое выражение в базисе И, ИЛИ, НЕ: $F = (clsd) \vee (\bar{a}bl\bar{c}ld) \vee (a\bar{c}ld) \vee (albcl\bar{d})$

4. Кирилл устроился работать помощником системного администратора в IT-отдел Университета. В первый же день работы Кирилл получил ответственное задание. Отдел информационных технологий Университета использует диапазон IP-адресов 192.168.64.0/22, в котором настроено адресное пространство сетей филиалов из 6 подсетей. Данные по распределению адресного пространства частично известны. При этом адресное пространство использовано максимально эффективно.

Кирилл обладает следующими знаниями:

- одному устройству должен соответствовать один IP-адрес вида XXX.XXX.XXX.XXX, при этом XXX – число в диапазоне [0:255], называемое октетом;
- устройства внутри одной подсети должны беспрепятственно взаимодействовать друг с другом и должны группироваться по их расположению;
- выделенный размер подсети содержит минимально необходимое количество адресов;
- подсети большего размера должны находиться в начале адресного диапазона сети;
- в каждом сегменте сети первый адрес зарезервирован под идентификатор подсети, а последний – под широковещательную рассылку;
- выделенный размер подсети записывается как $2^n - 2$, где n – минимальное количество бит, которое может содержать все номера адресов подсети.



Название подсети	Адрес сети	Выделенный размер	Префикс	Десятичная маска	Диапазон доступных адресов	Широковещательный адрес
Главный корпус		510				
Инженерный корпус					192.168.66.1 - ...	
Учебный центр №2	192.168.67.0		/25			
Библиотека						192.168.67.191
Пост охраны				255.255.255.224		
IT-отдел					... - 192.168.67.254	

Заполнение информации вызвало у помощника системного администратора затруднения. Необходимо помочь Кириллу заполнить таблицу недостающими данными. (20 баллов)

Решение:

Название подсети	Адрес сети	Выделенный размер	Префикс	Десятичная маска	Диапазон доступных адресов	Широковещательный адрес
Главный корпус	192.168.64.0	510	/23	255.255.254.0	192.168.64.1 - 192.168.65.254	192.168.65.255
Инженерный корпус	192.168.66.0	254	/24	255.255.255.0	192.168.66.1 - 192.168.66.254	192.168.66.255
Учебный центр №2	192.168.67.0	126	/25	255.255.255.128	192.168.67.1 - 192.168.67.126	192.168.67.127
Библиотека	192.168.67.128	62	/26	255.255.255.192	192.168.67.129 - 192.168.67.190	192.168.67.191
Пост охраны	192.168.67.192	30	/27	255.255.255.224	192.168.67.193 - 192.168.67.222	192.168.147.223
IT-отдел	192.168.67.224	30	/27	255.255.255.224	192.168.67.225 - 192.168.67.254	192.168.147.255

5. Администратору сообществ ВК Николаю необходимо определить, какие участники состоят в сообществе «Кубок веломарафонов Ленинградской области» (3353 участника) и одновременно НЕ состоят в сообществе «Грибы Ленинградской области» (74656 участников), и вывести на экран получившийся список. У администратора есть два упорядоченных списка участников каждой группы. В каждом списке участник представлен одним и тем же восьмизначным цифровым кодом. Без повторов внутри списка. Списки упорядочены по возрастанию кодов. Помогите Николаю составить алгоритм и программу для решения этой задачи, считая, что списки пригодны для непосредственной обработки выбранным языком программирования и уже загружены в память. Правильность решения докажете на простом примере. В решении **нельзя** использовать встроенные в язык операции, методы или свойства для работы со структурами данных, кроме как для определения длины структуры или добавления (удаления) элементов структуры. Код необходимо снабдить комментариями. (25 баллов)



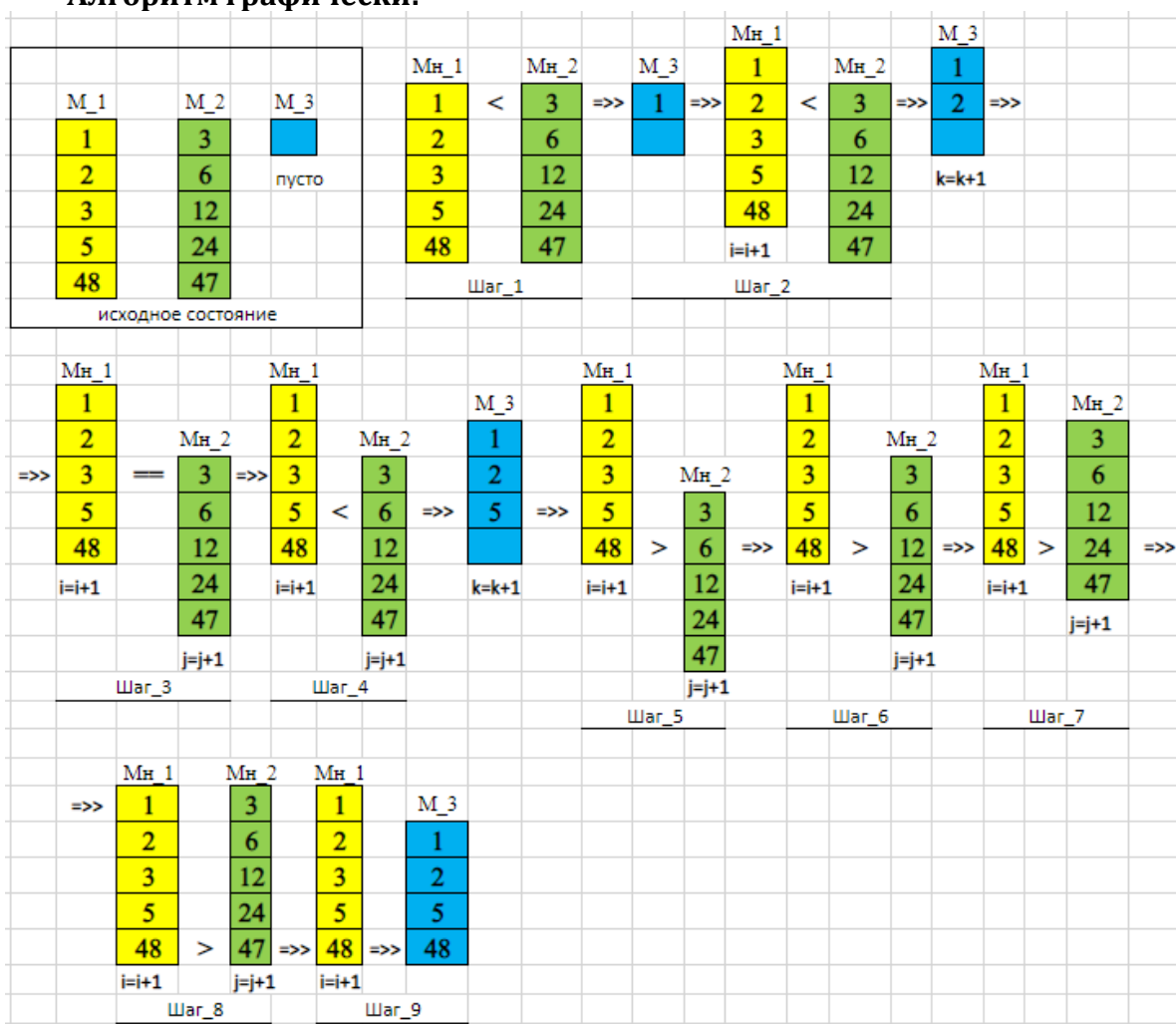
Решение:

Алгоритм решения задачи:

Задача на разность множеств. Разность двух упорядоченных множеств **A** и **B** – это множество с элементами **A**, не совпадающими с элементами **B**, без дублей. Сложность алгоритма $O(m+n)$, где **m** и **n** – длины входных упорядоченных множеств **A** и **B** соответственно.

Необходимо написать программу, аргументы которой – два упорядоченных массива для вычисления разности. Функция возвращает третий массив, содержащий элементы, которые есть только в первом массиве и нет во втором. Необходимо взять два исходных массива и пустой третий. Сделать и обнулить три счетчика индексов массивов: $i=0, j=0, k=0$. Сравниваем верхние (нулевые) элементы массивов. Если $M_{1i}==M_{2j}$, то: $i=i+1, j=j+1$ – поднимаем M_1 и M_2 вверх на один элемент. Если $M_{1i}<M_{2j}$, то: $M_{3k}==M_{1i}, i=i+1, k=k+1$ – поднимаем M_1 и M_3 вверх на один элемент. Иначе: $j=j+1$ – поднимаем M_2 вверх на один элемент. И так в цикле while пока $i<L1$ и $j<L2$, где L – длина массива. Если $\max(M_1)<\max(M_2)$, то на этом всё, все элементы перебрали. Если $\max(M_1)>\max(M_2)$, то не перебранными останутся элементы первого массива, превосходящие по величине любой элемент второго массива. Эти элементы включаем в третий массив (последний while в mathcad). Проверить индексацию массивов и правильность определения верхней границы массива в условии цикла while.

Алгоритм графически:





Пример решения в Mathcad:

$$M_1 := \begin{pmatrix} 1 \\ 2 \\ 3 \\ 5 \\ 48 \end{pmatrix} \quad M_2 := \begin{pmatrix} 3 \\ 6 \\ 12 \\ 24 \\ 47 \end{pmatrix} \quad \begin{array}{l} \text{length}(M_1) = 5 \\ \text{length}(M_2) = 5 \end{array}$$

```
diff_sort_arr(a,b) := (aL ← length(a) bL ← length(b))
(i ← 0 j ← 0 k ← 0)
while (i < aL) ∧ (j < bL)
  (i ← i + 1 j ← j + 1) if ai = bj
  otherwise
    if ai < bj
      ck ← ai
      (i ← i + 1 k ← k + 1)
    j ← j + 1 otherwise
    "Если a[i]>b[j] то счетчик i=stop"
"Прогон ниже обрабатывает случай a[i]>b[j]"
while i < aL
  (ck ← ai "Хвост массива a для a[i]>b[j]" )
  (i ← i + 1 k ← k + 1)
c
```

$$\text{diff_sort_arr}(M_1, M_2) = \begin{pmatrix} 1 \\ 2 \\ 5 \\ 48 \end{pmatrix}$$



Пример решения на Python:

```
*Гранит_2021
File Edit Format Run Options Window Help
23 ##Задача №5 - Пересечение множеств
24 ##Дано два массива в виде списков
25 aList=[1,2,3,5,48]
26 bList=[3,6,12,24,47]
27 cList=[]##Пустой массив
28 #len(aList) - длина массива
29 i=0##Счетчики
30 j=0
31 k=0
32 while i<len(aList) and j<len(bList):
33     if aList[i]==bList[j]:
34
35         i+=1##Увеличение счетчика на 1
36         j+=1
37     elif aList[i]<bList[j]:
38         cList.append(aList[i])##Добавление элемента в конец списка
39         i+=1
40         k+=1
41     else:
42         j+=1
43 while i<len(aList):
44     cList.append(aList[i])##Добавление элемента в конец списка
45     i+=1
46     k+=1
47 print(cList)

Python 3.8.1 Shell
File Edit Shell Debug Options Window Help
Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 22:39:24) [MSC v.1916
 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more informati
on.
>>>
= RESTART: D:\Users\kosarev_ov\Downloads\Гранит_2021_2_этап_Косарев_Э
адача_5.py
[1, 2, 5, 48]
>>> |

Ln: 6 Col: 4
```




Пример решения на PascalABC.NET:

```
PascalABC.NET 3.7.2
Файл  Правка  Вид  Программа  Сервис  Модули  Помощь
[Icons]
•Гранит_2021
1  program Zadacha_5; //Необязательный заголовок
2  //Задача на разность множеств
3  //В языке есть встроенные множества "set"
4  //и метод a.except(b) разности множеств
5  //По условиям задачи так делать нельзя.
6
7  //Динамические массивы с данными
8  var
9  a: array of integer := (1, 2, 3, 5, 48);
10 b: array of integer := (3, 6, 12, 24, 47);
11 //Выходной массив. изначально пустой
12 c: array of integer;
13 i, j, k: integer; //Счётчики=0 по умолчанию
14
15 begin
16 while (i < a.Length) and (j < b.Length) do
17     if a[i] = b[j] then //Ветка true
18         begin
19             i := i + 1;
20             j := j + 1;
21         end
22     else //Ветка false
23         if a[i] < b[j] then
24             begin
25                 SetLength(c, k + 1);
26                 c[k] := a[i];
27                 i := i + 1; k := k + 1;
28             end
29         else j := j + 1;
30 while (i < a.Length) do
31     begin
32         SetLength(c, k + 1);
33         c[k] := a[i];
34         i := i + 1; k := k + 1;
35     end;
36 writeln(c);
37
38 //А теперь с помощью встроенного метода
39 a.Except(b).Println;
40 end.
```

Окно вывода

```
[1, 2, 5, 48]
1 2 5 48
```

Ответ

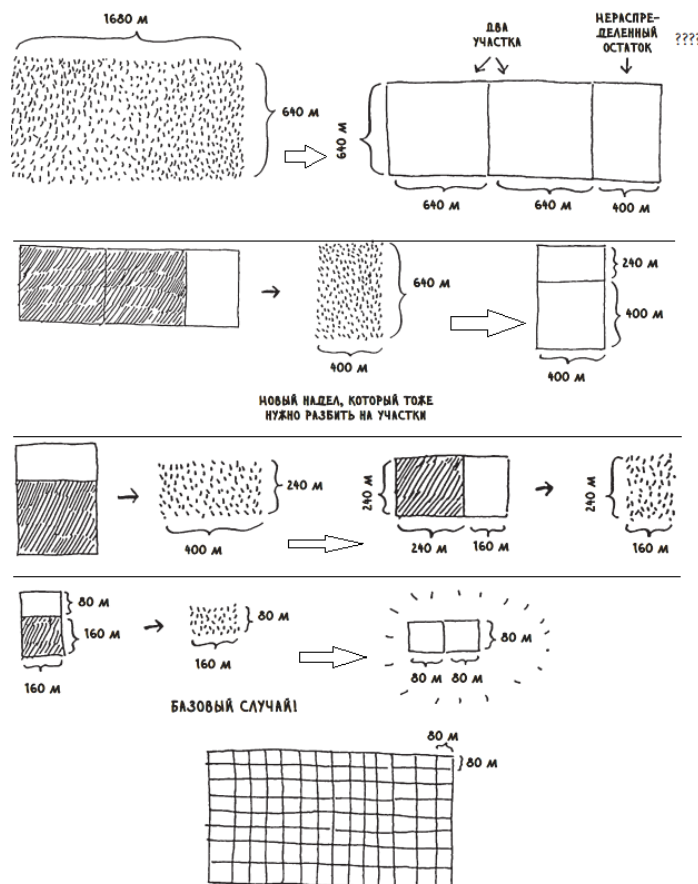


6. Оператору фрезерного станка необходимо выточить в алюминиевой заготовке прямоугольный паз размером 168х64 мм и глубиной 5 мм. Если использовать для этого фрезу малого диаметра, то можно сделать такой паз сразу очень точным, с ровными краями. Но это займет очень много времени и приведет к большому износу дорогостоящей фрезы. Оператор решил сначала высверлить основной объем паза обычным сверлом одного большого диаметра, а затем выполнить финишную обработку фрезой. Определите шаг сетки сверления такой, чтобы количество сверлений было минимальным, и количество таких сверлений. Напишите алгоритм и универсальную программу, которая на основе размеров прямоугольного участка вычисляет и выводит в качестве ответа размер шага сетки сверления и количество сверлений. Правильность решения докажите поэтапно вручную по коду. Код необходимо снабдить комментариями. **(25 баллов)**

Решение:

Алгоритм решения задачи:

Задача на использование рекурсии и алгоритма «Разделяй и властвуй». Есть некоторая поверхность прямоугольной формы с заданными сторонами. Ее необходимо разбить на минимальное количество одинаковых квадратов. Соответственно площадь квадратов должна быть максимальной. Решается с помощью «Алгоритма Евклида». Суть алгоритма: определить базовый случай и применить рекурсию. Базовый случай: соотношение сторон участка 2:1. Графическая иллюстрация ниже. Прямоугольник необходимо разбить на минимальное число квадратов. Сверло ставится в центр квадрата. Шаг сетки - расстояние между центрами квадратов = диаметр сверла = сторона квадрата. Количество сверлений = число квадратов. Ответ: сторона квадрата = 8 мм. Проверка поэтапно на рисунке с кодом. Проверка по сторонам: $168:8=21$; $64:8=8$; $21 \times 8=168$ сверлений.





Пример решения на Python:

```
Гранит_2021 Python 3.8.2 Shell
File Edit Format Run Options Window Help File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
== RESTART: C:\Users\User\Downloads\Гранит_2021_2_этап_Косарев_Задача_6_1_1.py ==
>>>
a= 168
b= 64
a= 64
b= 40
a= 40
b= 24
a= 24
b= 16
a= 16
b= 8
Сторона квадрата = 8
Ответ = 168.0
>>> |
Ln: 19 Col: 4
Ln: 31 Col: 0
```

Пример решения на PascalABC.NET:

```
Гранит_2021
1 program Zadacha_6; //Необязательный заголовок
2 //Задача на определение максимального квадрата,
3 //которым можно заполнить прямоугольник.
4 var //Переменные для хранения сторон прямоугольника
5 //a,b - стороны участка, m - остаток от деления mod
6 a1,b1,L: integer;
7 //рекурсивная функция
8 function kvadrat(a,b: integer):integer;
9 var m: integer;
10 begin //если указали стороны в неправильном порядке
11 println('a=',a); //для контроля хода решения
12 println('b=',b);
13 //базовый случай, отношение сторон 2/1 для
14 //подобранных примеров
15 if (a mod b)=0 then //это и есть базовый случай
16 begin
17 Result:=b; //возвращаемое значение
18 Println('Сторона квадрата = ', b);
19 end
20 else //базовый случай не достигнут
21 begin
22 m:=a mod b;
23 Result:=kvadrat(b,m); //запускаем рекурсию
24 end;
25 end;
26
27 begin //основная программа
28 a1:=168; b1:=64;
29 L:=kvadrat(a1,b1); //вызов рекурсивной функции
30 Println('Ответ = ', (a1/L)*(b1/L));
31 end.
```

```
Окно вывода
a= 168
b= 64
a= 64
b= 40
a= 40
b= 24
a= 24
b= 16
a= 16
b= 8
Сторона квадрата = 8
Ответ = 168
```

**ИНФОРМАТИКА****БИЛЕТ 2. ВАРИАНТ 2**

1. Пропуски в числовых массивах – кошмар перфекциониста Володи. Помогите ему заполнить ячейки, выделенные красным, желтым и зеленым цветами (или соответствующим узором). Дайте пояснения. Определить значения можно по следующим формулам:

Зеленая (узор «вертикальные линии»): =ЕСЛИ(R[-7]C20=1;R[-2]C[4]+R[-7]C[-2]*2;ЕСЛИ(R[2]C22=2;(R[-3]C[-6]/4+R[-8]C[1]/3)+3;(R[-8]C[-3]^2-R11C27*3)/6))

Красная (узор «клетка»): =ЕСЛИ(ИЛИ(R5C[-2]>R[4]C27;И(R6C<=R[-6]C21;R3C31>50;-R[4]C[1]<0));R[3]C29*(1-R[5]C[-4])/R[4]C[6]+1;R[-4]C[-3]+R[6]C[2]/R4C[1])

Желтая (узор «горизонтальные линии»): =ЕСЛИ(R[6]C[-2]<R3C21;R[4]C[5]/R[7]C[6]^2;ЕСЛИ(И(R[2]C[2]-R[4]C[-4]/R15C[5]<0;R[-1]C28>75);2;1))

Значения цветных ячеек (ячеек с узором) необходимо округлять до ближайшего целого числа.

	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
2	54	56	21	25	67	5	73	36	25	31	87	30	24	60	57
3	17	31	45	14	30	29	94	68	26	43	34	92	94	21	36
4	89	11	11	81	74	93	43	27	32	24	22	44	12	73	10
5	61	96	5	25	33		76	93	72	8	21	76	68	10	75
6	93	43	5	52	74	90	47	7	11	39	95	98	75	37	31
7	38	15	8	19	78	66	16	19	87	31	85	58	34	23	5
8	95	49	93	61	90	42	84	68	13	17	15	45	9	32	52
9	54	84	96	68	45	77	69		13	91	72	32	79	16	83
10	40	41	84	63	93	35	37	43	48	37	13	86	42	47	88
11	85	84	60	53	71	11	96	61	6	59	14	15	27	24	60
12	31	23	28	17	37	23	59	7		76	68	93	44	73	44
13	72	14	12	15	68	59	62	1	68	53	76	10	4	52	38
14	57	73	25	42	92	21	95	74	34	35	13	86	43	53	77
15	19	90	81	14	8	84	93	95	85	86	12	57	94	92	5

(8 баллов)

Ответ:

Зеленая (узор «вертикальные линии»): -3

Красная (узор «клетка»): -13

Желтая (узор «горизонтальные линии»): 1

Решение:

	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
2	54	56	21	25	67	5	73	36	25	31	87	30	24	60	57
3	17	31	45	14	30	29	94	68	26	43	34	92	94	21	36
4	89	11	11	81	74	93	43	27	32	24	22	44	12	73	10
5	61	96	5	25	33	1	76	93	72	8	21	76	68	10	75
6	93	43	5	52	74	90	47	7	11	39	95	98	75	37	31
7	38	15	8	19	78	66	16	19	87	31	85	58	34	23	5
8	95	49	93	61	90	42	84	67	13	17	15	45	9	32	52
9	54	84	96	68	45	77	69	-13	13	91	72	32	79	16	83
10	40	41	84	63	93	35	37	43	48	37	13	86	42	47	88
11	85	84	60	53	71	11	96	61	6	59	14	15	27	24	60
12	81	19	83	15	70	19	25	84	13	25	74	39	13	11	33
13	31	23	28	17	37	23	59	7	-3	76	68	93	44	73	44
14	72	14	12	15	68	59	62	1	68	53	76	10	4	52	38
15	57	73	25	42	92	21	95	74	34	35	13	86	43	53	77
16	19	90	81	14	8	84	93	95	85	86	12	57	94	92	5



2. Два гнома решили сыграть в игру «21 драгоценный камень». Они выложили в ряд 21 драгоценный камень. По условию игры можно поочередно брать 1, 2 или 3 камня за раз. Можно брать **любые** камни. Пропускать свой ход нельзя. **Проигрывает** тот, кто заберет **последние** камни или камень. Напишите алгоритм **выигрышной стратегии** для гнома, ходящего **вторым**. Поясните алгоритм графически. Напишите обобщенный алгоритм для любого нечетного количества камней. Укажите нижнюю границу применимости алгоритма. **(10 баллов)**

Решение:

Условие выигрыша Игрок_1 – $N\%(k+1) \neq 1$ + Стратегия_1 (см. вариант 1).

Условие выигрыша Игрок_2 – $N\%(k+1) = 1$ + Стратегия_2.

Стратегия: оставить противнику такой Остаток предметов, чтобы $Остаток\%(k+1) = 1$.

$21\%4 = 1$. Поэтому используется Стратегия_2. Всегда выигрывает Игрок_2 (игрок, делающий ход вторым). **Суть Стратегия_2:**

Ход_1: Игрок_1 (игрок, делающий ход первым) берет любое разрешенное количество предметов. Реализовать Стратегия_1 Игрок_2 не может, так как первая выигрышная комбинация – 17, а для этого нужно взять 4 предмета, что больше разрешенных 3-х.

Ход_2: Игрок_2 (игрок, делающий ход вторым) берет столько предметов, чтобы $Остаток\%(k+1) = 1$. Формула Остатка: $1 + i*(k+1)$, где $i = 1, 2, 3, 4, 5, \dots$. С этого момента реализуется Стратегия_1, но для Игрок_2. Получается, что теперь он становится первым. Игрок_1 обречен. Дальнейшие ходы повторяют Ход_1 и Ход_2.

Алгоритм для Игрок_2: Шаг 1 - $Остаток\%(k+1) = 1$. Повтор до Победа.

Нижняя граница – 5 предметов.

	Игрок_1 - ходит первым			Можно брать 1, 2 или 3 СМЕЖНЫХ предмета. Всего N предметов, N нечетное.																	k=3	
Вариант 2 Стратегия 2	Игрок_2 - ходит вторым																				3	
Победитель Игрок 2																						
Исходное состояние	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	5
Ход 1 - забрал 1,2 и 3	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	9
	Остаток=18, $18\%(3+1)=2$																	13				
Ход 2 - делаем Остаток=1				4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	17
	Остаток=17, $17\%(3+1)=1$																					
	И так далее до конца как в Вариант_1. В конце у Игрок_1 остается 1 предмет и он проигрывает.																					

3. На площадке размещения техники в открытом угольном карьере специальный светофор для карьерных самосвалов состоит из 4 секций, расположенных буквой Т (горизонтальная составляющая содержит три секции, а вертикальная две). Каждая секция может мигать оранжевым светом или быть выключенной. На работу светофора наложены следующие ограничения: а) должно гореть не менее двух секций; б) не может гореть только две секции в горизонтальном ряду. Нарисуйте **запрещенные** комбинации светофора. Составьте таблицу истинности работы светофора. Запишите логическую функцию, описывающую **запрещенные** комбинации светофора, в дизъюнктивной нормальной форме. Упростите полученное выражение, оставаясь в базисе И, ИЛИ, НЕ. **(12 баллов)**



Решение:

	Запрещенные комбинации				Разрешенные комбинации		
1				9			
2				10			
3				11			
4				12			
5				13			
6				14			
7				15			
8				16			

1. Обозначим буквами секции светофора: a, b, c – слева направо верхний ряд, d – секция в нижнем ряду. Составим таблицу истинности функции F .

N	a	b	c	d	F
0	0	0	0	0	1
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	0
4	0	1	0	0	1
5	0	1	0	1	0
6	0	1	1	0	1
7	0	1	1	1	0
8	1	0	0	0	1
9	1	0	0	1	0
10	1	0	1	0	1
11	1	0	1	1	0
12	1	1	0	0	1
13	1	1	0	1	0
14	1	1	1	0	0
15	1	1	1	1	0

Функция F истинна (равна 1) на номерах наборов 0,1,2,4,6,8,10 и 12.
 $F = (\bar{a}\bar{b}\bar{c}\bar{d}) \vee (\bar{a}\bar{b}\bar{c}d) \vee (\bar{a}\bar{b}c\bar{d}) \vee (\bar{a}\bar{b}cd) \vee (\bar{a}b\bar{c}\bar{d}) \vee (\bar{a}b\bar{c}d) \vee (\bar{a}bc\bar{d}) \vee (\bar{a}bcd)$

Упростим функцию. По ТИ видно, что $\bar{c}\bar{d}$ всегда дают $F=1$. Запишем новое выражение:

$$F = (\bar{c}\bar{d}) \vee (\bar{a}\bar{b}\bar{c}\bar{d}) \vee (\bar{a}\bar{b}c\bar{d}) \vee (\bar{a}b\bar{c}\bar{d}) \vee (\bar{a}bcd)$$



N	a	b	c	d	F
0	0	0	0	0	1
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	0
4	0	1	0	0	1
5	0	1	0	1	0
6	0	1	1	0	1
7	0	1	1	1	0
8	1	0	0	0	1
9	1	0	0	1	0
10	1	0	1	0	1
11	1	0	1	1	0
12	1	1	0	0	1
13	1	1	0	1	0
14	1	1	1	0	0
15	1	1	1	1	0

В полученном выражении можно минимизировать выражения в 3 и 4 скобках: $(\bar{a}\bar{b}lcl\bar{d})\vee(\bar{a}l\bar{b}lcl\bar{d})=(\bar{a}lcl\bar{d})$. Далее минимизировать нечего. Итоговое выражение в базисе И, ИЛИ, НЕ: $F = (\bar{c}l\bar{d})\vee(\bar{a}\bar{b}l\bar{c}ld)\vee(\bar{a}lcl\bar{d})\vee(al\bar{b}lcl\bar{d})$

4. Кирилл устроился работать помощником системного администратора в IT-отдел Университета. В первый же день работы Кирилл получил ответственное задание. Отдел информационных технологий Университета использует диапазон IP-адресов 192.168.68.0/22, в котором настроено адресное пространство сетей филиалов из 6 подсетей. Данные по распределению адресного пространства частично известны. При этом адресное пространство использовано максимально эффективно.

Кирилл обладает следующими знаниями:

- одному устройству должен соответствовать один IP-адрес вида XXX.XXX.XXX.XXX, при этом XXX – число в диапазоне [0:255], называемое октетом;
- устройства внутри одной подсети должны беспрепятственно взаимодействовать друг с другом и должны группироваться по их расположению;
- выделенный размер подсети содержит минимально необходимое количество адресов;
- подсети большего размера должны находиться в начале адресного диапазона сети;
- в каждом сегменте сети первый адрес зарезервирован под идентификатор подсети, а последний – под широковещательную рассылку;
- выделенный размер подсети записывается как $2^n - 2$, где n – минимальное количество бит, которое может содержать все номера адресов подсети.



Название подсети	Адрес сети	Выделенный размер	Префикс	Десятичная маска	Диапазон доступных адресов	Широковещательный адрес
Главный корпус		510				
Инженерный корпус					192.168.70.1 - ...	
Учебный центр №2	192.168.71.0		/25			
Библиотека						192.168.71.191
Пост охраны				255.255.255.224		
IT-отдел					... - 192.168.71.254	

Заполнение информации вызвало у помощника системного администратора затруднения. Необходимо помочь Кириллу заполнить таблицу недостающими данными. **(20 баллов)**

Решение:

Название подсети	Адрес сети	Выделенный размер	Префикс	Десятичная маска	Диапазон доступных адресов	Широковещательный адрес
Главный корпус	192.168.68.0	510	/23	255.255.254.0	192.168.68.1 - 192.168.69.254	192.168.69.255
Инженерный корпус	192.168.70.0	254	/24	255.255.255.0	192.168.70.1 - 192.168.70.254	192.168.70.255
Учебный центр №2	192.168.71.0	126	/25	255.255.255.128	192.168.71.1 - 192.168.71.126	192.168.71.127
Библиотека	192.168.71.128	62	/26	255.255.255.192	192.168.71.129 - 192.168.71.190	192.168.71.191
Пост охраны	192.168.71.192	30	/27	255.255.255.224	192.168.71.193 - 192.168.71.222	192.168.71.223
IT-отдел	192.168.71.224	30	/27	255.255.255.224	192.168.71.225 - 192.168.71.254	192.168.71.255

5. Директор РОНО попросил администратора базы данных принести распечатку со списком детей, посещающих секцию шахмат и при этом НЕ посещающих секцию настольного тенниса в районном Доме культуры. Администратор базы данных сохранил два упорядоченных списка детей, посещающих каждую секцию. В каждом списке участники представлены в виде уникального кода, состоящего из прописных букв латинского алфавита и цифр от 0 до 9. Внутри списка повторений и совпадений нет. Списки упорядочены по возрастанию. Помогите администратору составить алгоритм и программу для решения этой задачи, считая, что списки пригодны для непосредственной обработки выбранным языком программирования и уже загружены в память. Правильность решения докажете на простом примере. В решении **нельзя** использовать встроенные в язык операции, методы или свойства для работы со структурами данных, кроме как для определения длины структуры или добавления (удаления) элементов структуры. Код необходимо снабдить комментариями. **(25 баллов)**



Решение:

Алгоритм решения задачи:

Задача на разность множеств. Разность двух упорядоченных множеств **A** и **B** – это множество с элементами **A**, не совпадающими с элементами **B**, без дублей. Сложность алгоритма $O(m+n)$, где **m** и **n** – длины входных упорядоченных множеств **A** и **B** соответственно.

Необходимо написать программу, аргументы которой - два упорядоченных массива для вычисления разности. Функция возвращает третий массив, содержащий элементы, которые есть только в первом массиве и нет во втором. Необходимо взять два исходных массива и пустой третий. Сделать и обнулить три счетчика индексов массивов: $i=0, j=0, k=0$. Сравниваем верхние (нулевые) элементы массивов. Если $M_{1i}=M_{2j}$, то: $i=i+1, j=j+1$ – поднимаем M_1 и M_2 вверх на один элемент. Если $M_{1i}<M_{2j}$, то: $M_{3k}=M_{1i}, i=i+1, k=k+1$ – поднимаем M_1 и M_3 вверх на один элемент. Иначе: $j=j+1$ – поднимаем M_2 вверх на один элемент. И так в цикле while пока $i<L1$ и $j<L2$, где L – длина массива. Если $\max(M_1)<\max(M_2)$, то на этом всё, все элементы перебрали. Если $\max(M_1)>\max(M_2)$, то не перебранными останутся элементы первого массива, превосходящие по величине любой элемент второго массива. Эти элементы включаем в третий массив. Проверить индексацию массивов и правильность определения верхней границы массива в условии цикла while.

Пример решения на PascalABC.NET:

```
1 program Zadacha_5; //Необязательный заголовок
2 //Задача на разность множеств
3 //В языке есть встроенные множества "set"
4 //и метод a.экспорт(b) разности множеств
5 //По условиям задачи так делать нельзя.
6
7 //Динамические массивы с данными
8 var
9   a: array of string := ('aa2', 'bb', 'cc', 'ddl', 'ee');
10  b: array of string := ('aa2', 'ddl', 'de', 'ed', 'ff');
11 //Выходной массив, изначально пустой
12 c: array of string;
13 i, j, k: integer; //Счётчики=0 по умолчанию
14
15 begin
16   while (i < a.Length) and (j < b.Length) do
17     if a[i] = b[j] then //Ветка true
18       begin
19         i := i + 1;
20         j := j + 1;
21       end
22     else //Ветка false
23       if a[i] < b[j] then
24         begin
25           SetLength(c, k + 1);
26           c[k] := a[i];
27           i := i + 1; k := k + 1;
28         end
29       else j := j + 1;
30   while (i < a.Length) do
31     begin
32       SetLength(c, k + 1);
33       c[k] := a[i];
34       i := i + 1; k := k + 1;
35     end;
36   writeln(c);
37
38   //А теперь с помощью встроенного метода
39   a.Экспорт(b).Println;
40 end.
```

Окно вывода

```
[bb, cc, ee]
bb cc ee
```

← Ответ



6. Василий Иванович решил заняться разведением коз. Для этого он взял в аренду участок луга 36x14 м. для выпаса коз. Чтобы не следить за козами, Василий Иванович решил вбить колышек и привязать к нему козу. Пускай ходит по кругу и щиплет травку. Сколько коз сможет разместить таким образом на участке Василий Иванович с учетом того, чтобы каждой козе досталось одинаковое количество травы и участок был максимально большим? Напишите алгоритм и универсальную программу, которая на основе размеров участка луга вычисляет и выводит в качестве ответа количество мест для выпаса коз. Правильность решения докажете поэтапно вручную по коду. Код необходимо снабдить комментариями. **(25 баллов)**

Решение:

Алгоритм решения задачи:

Задача на использование рекурсии и алгоритма «Разделяй и властвуй». Есть некоторая поверхность прямоугольной формы с заданными сторонами. Ее необходимо разбить на минимальное количество одинаковых квадратов. Соответственно площадь квадратов должна быть максимальной. Решается с помощью «Алгоритма Евклида». Суть алгоритма: определить базовый случай и применить рекурсию. Базовый случай: соотношение сторон участка 2:1. Графическая иллюстрация см. вариант 1. Прямоугольник необходимо разбить на минимальное число квадратов. Колышек вбивается в центр квадрата. Радиус привязи = 0.5xсторона квадрата. Количество коз = число квадратов. Ответ: сторона квадрата = 2 м. Проверка поэтапно на рисунке с кодом. Проверка по сторонам: $36:2=18$; $14:2=7$; $18 \times 7=126$ коз.

Пример решения на PascalABC.NET:

```
•Гранит_2021
1 program Zadacha_6; //Необязательный заголовок
2 //Задача на определение максимального квадрата,
3 //которым можно заполнить прямоугольник.
4 var //Переменные для хранения сторон прямоугольника
5 //a,b - стороны участка, m - остаток от деления mod
6   al,bl,L: integer;
7 //рекурсивная функция
8 function kvadrat(a,b: integer):integer;
9 var m: integer;
10 begin //если указали стороны в неправильном порядке
11   println('a=',a); //для контроля хода решения
12   println('b=',b);
13   //базовый случай, отношение сторон 2/1 для
14   //подобранных примеров
15   if (a mod b)=0 then //это и есть базовый случай
16     begin
17       Result:=b; //возвращаемое значение
18       Println('Сторона квадрата = ', b);
19     end
20   else //базовый случай не достигнут
21     begin
22       m:=a mod b;
23       Result:=kvadrat(b,m); //запускаем рекурсию
24     end;
25 end;
26
27 begin //основная программа
28   al:=36; bl:=14;
29   L:=kvadrat(al,bl); //вызов рекурсивной функции
30   Println('Ответ = ', (al/L)*(bl/L));
31 end.
```

Окно вывода

```
a= 36
b= 14
a= 14
b= 8
a= 8
b= 6
a= 6
b= 2
Сторона квадрата = 2
Ответ = 126
```

**ИНФОРМАТИКА****БИЛЕТ 2. ВАРИАНТ 3**

1. Пропуски в числовых массивах – кошмар перфекциониста Володи. Помогите ему заполнить ячейки, выделенные красным, желтым и зеленым цветами (или соответствующим узором). Определить значения можно по следующим формулам:

Зеленая (узор «вертикальные линии»): =ЕСЛИ(R[8]C39=1;R[11]C[4]+R[12]C[2]*2;ЕСЛИ(R[6]C39=2;(R[9]C/4+R[8]C[4]/3)+3;(R[2]C^2-R10C37*3)/6))

Красная (узор «клетка»): =ЕСЛИ(ИЛИ(R6C>R[-5]C42;И(R[-1]C[-4]=R[-1]C46;R11C48>50;-R[-5]C<0));R[-4]C47*(1-R[5]C[-4])/R[4]C[-4]+1;R[-4]C[-3]+R[6]C[2]/R4C[1])

Желтая (узор «горизонтальные линии»): =ЕСЛИ(R[5]C[4]<R3C46;R[-3]C[-6]/R[3]C[7]*2;ЕСЛИ(И(RC[4]-R[7]C/R4C[-1]<0;R[2]C38>75);2;1))

Значения цветных ячеек (ячеек с узором) необходимо округлять до ближайшего целого числа.

	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49
2	76	93	74	77	76	76	64	84	52	49	3	45	31	42	63
3	18	95	1	49	52	6	31	57	27	68	13	94	87	40	61
4	60	17	70	65	3	56	94	86	5	53	75	73	75	86	
5	68	63	21	14	28	70	77	77	88	9	33	61	21	65	72
6	23	87	64	85	45	28	26	67	100	96	25	21	20	59	
7	24	50	75	35	100	83	41	12	70	21	10	41	40	80	36
8	13	75	24	12	21	47	27	30	39	76	98	56	58	82	31
9	46	73	21	30	15	65	86	72	72	100	89	9	98	31	
10	34	94	68	8	92	44	37	94	65	79	3	80	39	20	73
11	35	16	26	37	30	41	87	72	81	1	99	74	51	64	72
12	22	34	53	9	33	91	21	10	50	16	25	3	13	9	39
13	53	50	33	50	31	23	85	84	80	60	97	80	99	1	79
14	74	30	90	43	72	2	10	58	93	88	27	60	34	55	58
15	84	76	36	66	80	83	96	98	47	75	27	13	79	25	22
16	83	38	49	10	37	57	89	7	56	36	51	55	42	50	23

(8 баллов)

Ответ:

Зеленая (узор «вертикальные линии»): -28

Красная (узор «клетка»): -17

Желтая (узор «горизонтальные линии»): 6

Решение:

	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49
2	76	93	74	77	76	76	64	84	52	49	3	45	31	42	63
3	18	95	1	49	52	6	31	57	27	68	13	94	87	40	61
4	60	17	70	65	3	56	94	-28	86	5	53	75	73	75	86
5	68	63	21	14	28	70	77	77	88	9	33	61	21	65	72
6	23	87	64	85	45	28	26	6	67	100	96	25	21	20	59
7	24	50	75	35	100	83	41	12	70	21	10	41	40	80	36
8	13	75	24	12	21	47	27	30	39	76	98	56	58	82	31
9	46	73	21	30	15	65	86	-17	72	72	100	89	9	98	31
10	34	94	68	8	92	44	37	94	65	79	3	80	39	20	73
11	35	16	26	37	30	41	87	70	81	1	99	74	51	64	72
12	22	34	53	9	33	91	21	10	50	16	25	3	12	9	39
13	53	50	33	50	31	23	85	84	80	60	97	80	99	1	79
14	74	30	90	43	72	2	10	58	93	88	27	60	34	55	58
15	84	76	36	66	80	83	96	98	47	75	27	13	79	25	22
16	83	38	49	10	37	57	83	7	56	36	51	55	42	50	23



2. Два гнома решили сыграть в игру «20 драгоценных камней». Они выложили в ряд 20 драгоценных камней. По условию игры можно поочередно брать 1, 2, 3 или 4 камня за раз. Можно брать **любые** камни. Пропускать свой ход нельзя. **Проигрывает** тот, кто заберет **последние** камни или камень. Напишите алгоритм **выигрышной стратегии** для гнома, ходящего **первым**. Поясните алгоритм графически. Напишите обобщенный алгоритм для любого четного количества камней. Укажите нижнюю границу применимости алгоритма. (10 баллов)

Решение:

Условие выигрыша Игрок_1 – $N \%(k+1) \neq 1$ + Стратегия_1.

Условие выигрыша Игрок_2 – $N \%(k+1) = 1$ + Стратегия_2.

Стратегия: оставить противнику такой Остаток предметов, чтобы $Остаток \%(k+1) = 1$.

$4+1=5$. $20\%5=0 \neq 1$. Поэтому используется Стратегия_1. Всегда выигрывает Игрок_1 (игрок, делающий ход первым).

Суть Стратегия_1:

Ход_1: Взять столько предметов, чтобы $Остаток \%(k+1) = 1$. Формула Остатка: $1 + i \cdot (k+1)$, где $i = 1, 2, 3, 4, 5 \dots$. Берем 4, остаток 16, $16\%5=1$.

Ход_2: Игрок_2 (игрок, делающий ход вторым) берет любое разрешенное количество предметов. Реализовать Стратегия_1 Игрок_2 не может, так как следующая выигрышная комбинация – 11, а для этого нужно взять 5 предметов, что больше разрешенных 4-х. Игрок_2 обречен. Дальнейшие ходы повторяют Ход_1 и Ход_2.

Алгоритм для Игрок_1 (обобщенный): Шаг 1 – $Остаток \%(k+1) = 1$. Повтор до Победа.

Нижняя граница – 2 предмета.

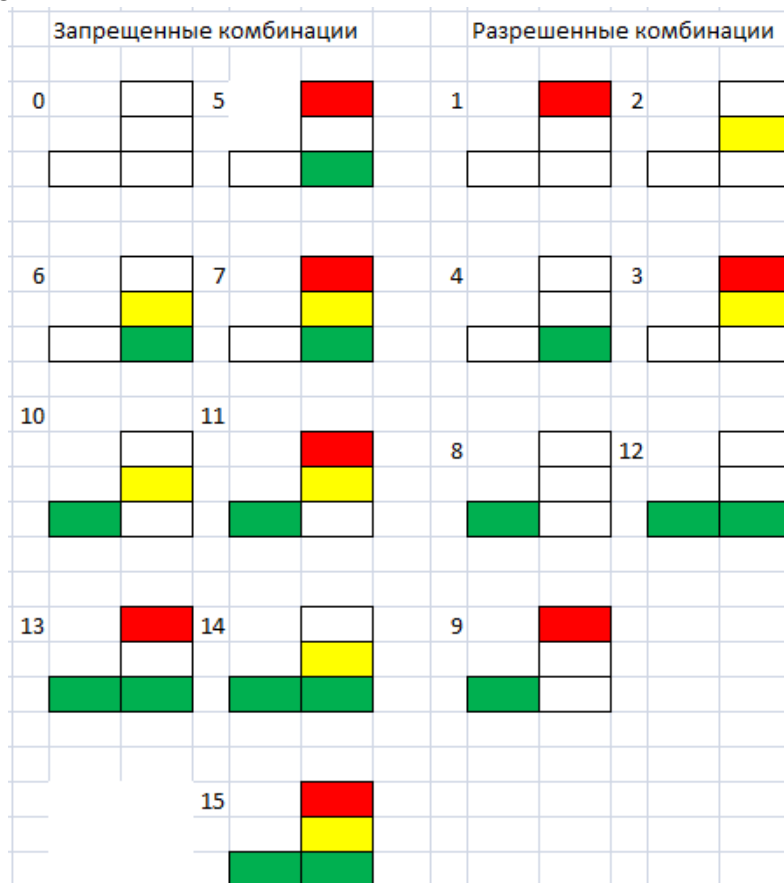
	Игрок_1 - ходит первым																Игрок_2 - ходит вторым				k=4	
Вариант 3 Стратегия 1	Игрок_1 - ходит первым																Игрок_2 - ходит вторым					
Победитель Игрок 1																					6	
Исходное состояние	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	11	16
Ход 1 - забрал 1,2,3 и 4	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	Остаток=16, $16\%(4+1)=1$	
Ход 2 - любой набор из 1,2,3 или 4-х предметов					5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	Остаток=12, $12\%(4+1)=2$	
Ход 3 - делаем остаток 1									9	10	11	12	13	14	15	16	17	18	19	20	Остаток=11, $11\%(4+1)=1$	
Ход 2 - любой набор из 1,2,3 или 4-х предметов										10	11	12	13	14	15	16	17	18	19	20	Остаток=7, $7\%(4+1)=2$	
Ход 5 - делаем остаток 1														14	15	16	17	18	19	20	Остаток=6, $6\%(4+1)=1$	
Независимо от хода Игрок_2 на поле останется $1 < N < 4$ предметов. А значит Игрок_1 сможет за один ход оставить 1 предмет и выиграть.																						

3. Светофор на пересечении четырехполосной дороги с шестиполосной имеет четыре секции: красную, желтую, зеленую (прямо), дополнительную зеленую (вправо по диагонали). Четырехполосная дорога имеет ограничение скоростного режима 60 км/ч, а шестиполосная дорога – 80 км/ч. Секции расположены в горизонтальный ряд слева направо в порядке упоминания. Возможны следующие комбинации работающих секций: только красная, только желтая, только зеленая, одновременно красная и желтая, только зеленая дополнительная, одновременно



зеленая и зеленая дополнительная, одновременно красная и зеленая дополнительная. Нарисуйте разрешенные комбинации светофора. Составьте таблицу истинности работы светофора. Запишите логическую функцию, описывающую разрешенные комбинации светофора, в дизъюнктивной нормальной форме. Упростите полученное выражение, оставаясь в базисе И, ИЛИ, НЕ. (12 баллов)

Решение:



1. Обозначим буквами секции светофора: a, b, c – сверху вниз вертикальный ряд от красного до зеленого, d – дополнительная зеленая секция. Составим таблицу истинности функции F .

N	a	b	c	d	F
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	1
4	0	1	0	0	1
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	0
8	1	0	0	0	1
9	1	0	0	1	1
10	1	0	1	0	0
11	1	0	1	1	0
12	1	1	0	0	1
13	1	1	0	1	0
14	1	1	1	0	0
15	1	1	1	1	0



Функция F истинна (равна 1) на номерах наборов 1,2,3,4,8,9 и 12.

$$F = (\bar{a}\bar{b}\bar{c}ld) \vee (\bar{a}\bar{b}lcl\bar{d}) \vee (\bar{a}\bar{b}lclsd) \vee (\bar{a}b\bar{c}l\bar{d}) \vee (a\bar{b}l\bar{c}l\bar{d}) \vee (a\bar{b}l\bar{c}ld) \vee (a\bar{b}l\bar{c}l\bar{d})$$

Упростим функцию. По ТИ видно, что в наборах 2-3 и 8-9 можно попарно сократить переменную d , а в наборах 4-12 сократить переменную a . Запишем новое выражение:

$F = (\bar{a}\bar{b}l\bar{c}ld) \vee (\bar{a}\bar{b}l\bar{c}l) \vee (b\bar{c}l\bar{d}) \vee (a\bar{b}l\bar{c})$. Дальше минимизировать нечего. Это и есть итоговое выражение в базе И, ИЛИ, НЕ.

N	a	b	c	d	F
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	1
4	0	1	0	0	1
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	0
8	1	0	0	0	1
9	1	0	0	1	1
10	1	0	1	0	0
11	1	0	1	1	0
12	1	1	0	0	1
13	1	1	0	1	0
14	1	1	1	0	0
15	1	1	1	1	0

4. Кирилл устроился работать помощником системного администратора в IT-отдел Университета. В первый же день работы Кирилл получил ответственное задание. Отдел информационных технологий Университета использует диапазон IP-адресов 192.168.72.0/22, в котором настроено адресное пространство сетей филиалов из 6 подсетей. Данные по распределению адресного пространства частично известны. При этом адресное пространство использовано максимально эффективно.

Кирилл обладает следующими знаниями:

- одному устройству должен соответствовать один IP-адрес вида XXX.XXX.XXX.XXX, при этом XXX – число в диапазоне [0:255], называемое октетом;
- устройства внутри одной подсети должны беспрепятственно взаимодействовать друг с другом и должны группироваться по их расположению;
- выделенный размер подсети содержит минимально необходимое количество адресов;
- подсети большего размера должны находиться в начале адресного диапазона сети;
- в каждом сегменте сети первый адрес зарезервирован под идентификатор подсети, а последний – под широковещательную рассылку;
- выделенный размер подсети записывается как $2^n - 2$, где n – минимальное количество бит, которое может содержать все номера адресов подсети.



Название подсети	Адрес сети	Выделенный размер	Префикс	Десятичная маска	Диапазон доступных адресов	Широковещательный адрес
Главный корпус		510				
Инженерный корпус					192.168.74.1 - ...	
Учебный центр №2	192.168.75.0		/25			
Библиотека						192.168.75.191
Пост охраны				255.255.255.224		
IT-отдел					... - 192.168.75.254	

Заполнение информации вызвало у помощника системного администратора затруднения. Необходимо помочь Кириллу заполнить таблицу недостающими данными. **(20 баллов)**

Решение

Название подсети	Адрес сети	Выделенный размер	Префикс	Десятичная маска	Диапазон доступных адресов	Широковещательный адрес
Главный корпус	192.168.72.0	510	/23	255.255.254.0	192.168.72.1 - 192.168.73.254	192.168.73.255
Инженерный корпус	192.168.74.0	254	/24	255.255.255.0	192.168.74.1 - 192.168.74.254	192.168.74.255
Учебный центр №2	192.168.75.0	126	/25	255.255.255.128	192.168.75.1 - 192.168.75.126	192.168.75.127
Библиотека	192.168.75.128	62	/26	255.255.255.192	192.168.75.129 - 192.168.75.190	192.168.75.191
Пост охраны	192.168.75.192	30	/27	255.255.255.224	192.168.75.193 - 192.168.75.222	192.168.75.223
IT-отдел	192.168.75.224	30	/27	255.255.255.224	192.168.75.225 - 192.168.75.254	192.168.75.255

5. Директор самой большой школы города на 2000 учеников попросил учителя информатики выяснить список учеников школы, которые зарегистрировалось для участия в городской олимпиаде по информатике, но не приняли в ней участие. И дал учителю на флешке два файла: в одном список учеников школы, прошедших регистрацию для участия, в другом список участников олимпиады. Оба списка представляют собой упорядоченный набор уникальных кодов учеников. Каждый ученик закодирован в списке заглавными буквами русского алфавита и цифрой от 1 до 8. Помогите учителю информатики составить алгоритм и программу для решения этой задачи, считая, что списки пригодны для непосредственной обработки выбранным языком программирования и уже загружены в память. Правильность решения докажете на простом примере. В решении **нельзя** использовать встроенные в язык операции, методы или свойства для работы со структурами данных, кроме как для определения длины структуры или добавления (удаления) элементов структуры. Код необходимо снабдить комментариями. **(25 баллов)**



Решение:

Алгоритм решения задачи:

Задача на разность множеств. Разность двух упорядоченных множеств **A** и **B** – это множество с элементами **A**, не совпадающими с элементами **B**, без дублей. Сложность алгоритма $O(m+n)$, где **m** и **n** – длины входных упорядоченных множеств **A** и **B** соответственно.

Необходимо написать программу, аргументы которой - два упорядоченных массива для вычисления разности. Функция возвращает третий массив, содержащий элементы, которые есть только в первом массиве и нет во втором. Необходимо взять два исходных массива и пустой третий. Сделать и обнулить три счетчика индексов массивов: $i=0, j=0, k=0$. Сравниваем верхние (нулевые) элементы массивов. Если $M_{1i}==M_{2j}$, то: $i=i+1, j=j+1$ – поднимаем M_1 и M_2 вверх на один элемент. Если $M_{1i}<M_{2j}$, то: $M_{3k}:=M_{1i}, i=i+1, k=k+1$ – поднимаем M_1 и M_3 вверх на один элемент. Иначе: $j=j+1$ – поднимаем M_2 вверх на один элемент. И так в цикле while пока $i<L1$ и $j<L2$, где L – длина массива. Если $\max(M_1)<\max(M_2)$, то на этом всё, все элементы перебрали. Если $\max(M_1)>\max(M_2)$, то не перебранными останутся элементы первого массива, превосходящие по величине любой элемент второго массива. Эти элементы включаем в третий массив. Проверить индексацию массивов и правильность определения верхней границы массива в условии цикла while.

Пример решения на PascalABC.NET:

```
1  програм Zadacha_5; //Необязательный заголовок
2  //Задача на разность множеств
3  //В языке есть встроенные множества "set"
4  //и метод a.except(b) разности множеств
5  //По условиям задачи так делать нельзя.
6
7  //Динамические массивы с данными
8  var
9  a: array of string := ('AA2', 'BB', 'BB', 'ГГ1', 'ДД');
10 b: array of string := ('AA2', 'ГГ1', 'ДЕ', 'ЕД', 'ЖЖ');
11 //Выходной массив. изначально пустой
12 c: array of string;
13 i, j, k: integer; //Счётчики=0 по умолчанию
14
15 begin
16 while (i < a.Length) and (j < b.Length) do
17   if a[i] = b[j] then //Ветка true
18     begin
19       i := i + 1;
20       j := j + 1;
21     end
22   else //Ветка false
23     if a[i] < b[j] then
24       begin
25         SetLength(c, k + 1);
26         c[k] := a[i];
27         i := i + 1; k := k + 1;
28       end
29     else j := j + 1;
30 while (i < a.Length) do
31   begin
32     SetLength(c, k + 1);
33     c[k] := a[i];
34     i := i + 1; k := k + 1;
35   end;
36 writeln(c);
37
38 //А теперь с помощью встроенного метода
39 a.Except(b).Println;
40 end.
```

Окно вывода
[BB, BB, DD]
BB BB DD

← Ответ



6. Не очень сообразительный, но хитрый пастух принес скорняку шкуру барана и попросил сшить из нее так много шапок, сколько скорняк сможет. Причем самых больших, какие только скорняк сможет сшить! И чтобы все шапки были одинаковые по размеру. Шапки скорняк шил из круглой заготовки, а шкуру пастух принес прямоугольной формы размером 120x90 см. Скорняк взял свой смартфон, открыл на нем Pascal N-IDE (а может это был Pydroid 3) и запустил программу NatCalc. Программу ему написал внук, у которого в школе был предмет «Информатика». Напишите алгоритм и универсальную программу, которая на основе размеров прямоугольной шкуры вычисляет и выводит в качестве ответа количество одинаковых шапок максимального размера и диаметр заготовки. Правильность решения докажите поэтапно вручную по коду. Код необходимо снабдить комментариями. (25 баллов)

Решение:

Алгоритм решения задачи

Задача на использование рекурсии и алгоритма «Разделяй и властвуй». Есть некоторая поверхность прямоугольной формы с заданными сторонами. Ее необходимо разбить на минимальное количество одинаковых квадратов. Соответственно площадь квадратов должна быть максимальной. Решается с помощью «Алгоритма Евклида». Суть алгоритма: определить базовый случай и применить рекурсию. Базовый случай: соотношение сторон участка 2:1. Графическая иллюстрация см. вариант 1. Прямоугольник необходимо разбить на минимальное число квадратов. Ответ: диаметр заготовки = сторона квадрата = 30 см. Проверка поэтапно на рисунке с кодом. Проверка по сторонам: $120:30=4$; $90:30=3$; $4 \times 3=12$ шапок.

Пример решения на PascalABC.NET:

```
•Гранит_2021
1 program Zadacha_6; //Необязательный заголовок
2 //Задача на определение максимального квадрата,
3 //которым можно заполнить прямоугольник.
4 var //Переменные для хранения сторон прямоугольника
5 //a,b - стороны участка, m - остаток от деления mod
6     a,b,L: integer;
7 //рекурсивная функция
8 function kvadrat(a,b: integer):integer;
9 var m: integer;
10 begin //если указали стороны в неправильном порядке
11     println('a=',a); //для контроля хода решения
12     println('b=',b);
13     //базовый случай, отношение сторон 2/1 для
14     //подобранных примеров
15     if (a mod b)=0 then //это и есть базовый случай
16     begin
17         Result:=b; //возвращаемое значение
18         Println('Сторона квадрата = ', b);
19     end
20     else //базовый случай не достигнут
21     begin
22         m:=a mod b;
23         Result:=kvadrat(b,m); //запускаем рекурсию
24     end;
25 end;
26
27 begin //основная программа
28     a:=120; b:=90;
29     L:=kvadrat(a,b); //вызов рекурсивной функции
30     Println('Ответ = ', (a/L) * (b/L));
31 end.
```

Окно вывода

```
a= 120
b= 90
a= 90
b= 30
Сторона квадрата = 30
Ответ = 12
```

**ИНФОРМАТИКА****БИЛЕТ 2. ВАРИАНТ 4**

1. Пропуски в числовых массивах – кошмар перфекциониста Володи. Помогите ему заполнить ячейки, выделенные красным, желтым и зеленым цветами (или соответствующим узором). Дайте пояснения. Определить значения можно по следующим формулам:

Зеленая (узор «вертикальные линии»): $=ЕСЛИ(R[-7]C58=1;R[2]C[-1]+R[3]C[-3]*2;ЕСЛИ(R[-4]C57=2;(RC[-5]/4+R[-1]C[-1])/3)+3;(R[2]C[-7]^2-R4C60*3)/6))$

Красная (узор «клетка»): $=ЕСЛИ(ИЛИ(R15C[-2]>R[4]C63;И(R6C<=R[-1]C55;R13C60>50;-R[4]C[5]<0));R[-2]C61*(1-R[5]C[-4])/R[4]C[-4]+1;R[-4]C[-3]+R[6]C[2]/R4C[1])$

Желтая (узор «горизонтальные линии»): $=ЕСЛИ(R[-3]C[2]<R5C64;R[-6]C[5]/R[-5]C[5]^2;ЕСЛИ(И(R[-8]C[2]-R[-1]C[-2])/R12C[7]<0;R[-12]C52>75);2;1))$

Значения цветных ячеек (ячеек с узором) необходимо округлять до ближайшего целого числа.

	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65
2	84	100	52	59	66	63	88	3	36	18	32	65	10	23	99
3	16	52	69	77	88	39	13	27	76	19	65	28	13	66	18
4	74	23	10	86	100	20	23	76	62	26	71	30	1	33	73
5	74	90	3	90	40	3	90	99	25	93	80	54	46	64	95
6	68	23	91	24	94	31	12	7	55	52	58	3	81	6	26
7	82	49	18	4	83	61	73	84	7	3	84	48	96	34	41
8	48	24	48	33	89	45	93	1	83	35	15	70	28	63	56
9	61	40	78	71	46	20	58	10	67	18	62	6	37	13	
10	44	56	2	64	61	51	51	76	68	76	1	41	50	60	99
11	42	16	21	61	29	62	86	1	13	34	96	95	8	67	100
12	90	65	33	81	35	4	79	24	2	48	42	5	32	21	16
13	36	4	25	91	27	38	62	65	63	31	47	19	7	3	
14	16	74	56	82	51	63	41	74	66	24	96	31	96	5	8
15	15	20	9	37	89	18	9	6	93	61	1	67	100	8	79
16	62	24	91	96	12	85	98	43	49	96	59	6	6	53	89

(8 баллов)**Ответ:**

Зеленая (узор «вертикальные линии»): 41

Красная (узор «клетка»): -74

Желтая (узор «горизонтальные линии»): 18

Решение:

	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65
2	84	100	52	59	66	63	88	3	36	18	32	65	10	23	99
3	16	52	69	77	88	39	13	27	76	19	65	28	13	66	18
4	74	23	10	86	100	20	23	76	62	26	71	30	1	33	73
5	74	90	3	90	40	3	90	99	25	93	80	54	46	64	95
6	68	23	91	24	97	31	12	7	55	52	58	3	81	6	26
7	82	49	18	4	83	61	73	84	7	3	84	48	96	34	41
8	48	24	48	33	89	45	93	1	83	35	15	70	28	63	56
9	61	40	78	71	46	20	58	-74	10	67	18	62	6	37	13
10	44	56	2	64	61	51	51	76	68	76	1	41	50	60	99
11	42	16	21	61	29	62	86	1	13	34	96	95	8	67	100
12	90	65	33	81	35	4	79	24	2	48	42	5	32	21	16
13	36	4	25	91	27	38	62	65	63	31	47	19	41	7	3
14	15	74	56	82	51	63	41	74	66	24	96	31	96	5	8
15	15	20	9	37	89	18	9	6	93	61	1	67	100	8	79
16	62	24	91	96	12	85	98	43	49	96	59	6	6	53	89



2. Два гнома решили сыграть в игру «21 драгоценный камень». Они выложили в ряд 21 драгоценный камень. По условию игры можно поочередно брать 1, 2, 3 или 4 камня за раз. Можно брать **любые** камни. Пропускать свой ход нельзя. **Проигрывает** тот, кто заберет **последние** камни или камень. Напишите алгоритм **выигрышной стратегии** для гнома, ходящего **вторым**. Поясните алгоритм графически. Напишите обобщенный алгоритм для любого нечетного количества камней. Укажите нижнюю границу применимости алгоритма. (10 баллов)

Решение:

Условие выигрыша Игрок_1 – $N \%(k+1) \neq 1$ + Стратегия_1.

Условие выигрыша Игрок_2 – $N \%(k+1) = 1$ + Стратегия_2.

Стратегия: оставить противнику такой Остаток предметов, чтобы $\text{Остаток} \%(k+1) = 1$.

$4+1=5$. $21 \%(5)=1$. Поэтому используется Стратегия_2. Всегда выигрывает Игрок_2 (игрок, делающий ход вторым). **Суть Стратегия_2:**

Ход_1: Игрок_1 (игрок, делающий ход первым) берет любое разрешенное количество предметов. Реализовать Стратегия_1 Игрок_2 не может, так как первая выигрышная комбинация - 16, а для этого нужно взять 5 предметов, что больше разрешенных 4-х.

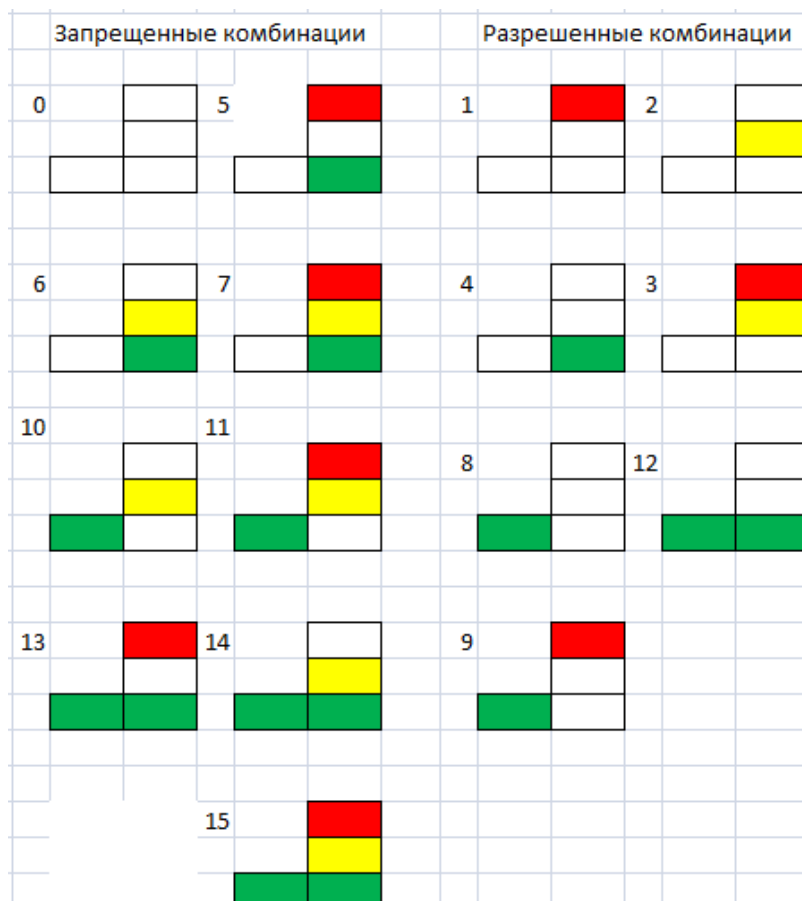
Ход_2: Игрок_2 (игрок, делающий ход вторым) берет столько предметов, чтобы $\text{Остаток} \%(k+1) = 1$. Формула Остатка: $1 + i \cdot (k+1)$, где $i = 1, 2, 3, 4, 5, \dots$. С этого момента реализуется Стратегия_1, но для Игрок_2. Получается, что теперь он становится первым. Игрок_1 обречен. Дальнейшие ходы повторяют Ход_1 и Ход_2. **Алгоритм** для Игрок_2 (обобщенный): Шаг 1 - $\text{Остаток} \%(k+1) = 1$. Повтор до Победа.

Нижняя граница – 6 предметов.

	Игрок_1 - ходит первым																					Можно брать 1, 2 или 3 СМЕЖНЫХ предмета. Всего N предметов, N нечетное.	k=4
Вариант 2 Стратегия 2	Игрок_2 - ходит вторым																						
Победитель Игрок 2																							
Исходное состояние	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	6	
Ход 1 - забрал 1,2,3 и 4	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	11	
	Остаток=17, $17 \%(4+1)=2$																					16	
Ход 2 - делаем Остаток=1					5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21		
	Остаток=16, $16 \%(4+1)=1$																						
	И так далее до конца как в Вариант_1. В конце у Игрок_1 остается 1 предмет и он проигрывает.																						

3. Светофор на пересечении четырехполосной дороги с шестиполосной имеет четыре секции: красную, желтую, зеленую (прямо), дополнительную зеленую (вправо по диагонали). Четырехполосная дорога имеет ограничение скоростного режима 60 км/ч, а шестиполосная дорога – 80 км/ч. Секции расположены в горизонтальный ряд слева направо в порядке упоминания. Возможны следующие комбинации работающих секций: только красная, только желтая, только зеленая, одновременно красная и желтая, только зеленая дополнительная, одновременно зеленая и зеленая дополнительная, одновременно красная и зеленая дополнительная. Нарисуйте **запрещенные** комбинации светофора. Составьте таблицу истинности работы светофора. Запишите логическую функцию, описывающую **запрещенные** комбинации светофора, в дизъюнктивной нормальной форме. Упростите полученное выражение, оставаясь в базисе И, ИЛИ, НЕ. (12 баллов)

Решение:



1. Обозначим буквами секции светофора: a, b, c – сверху вниз вертикальный ряд от красного до зеленого, d – дополнительная зеленая секция. Составим таблицу истинности функции F .

N	a	b	c	d	F
0	0	0	0	0	1
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	0
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	1
7	0	1	1	1	1
8	1	0	0	0	0
9	1	0	0	1	0
10	1	0	1	0	1
11	1	0	1	1	1
12	1	1	0	0	0
13	1	1	0	1	1
14	1	1	1	0	1
15	1	1	1	1	1

Функция F истинна (равна 1) на номерах наборов 0,5,6,7,10,11,13,14 и 15.
 $F = (\bar{a}\bar{b}\bar{c}\bar{d}) \vee (\bar{a}b\bar{c}\bar{d}) \vee (\bar{a}b\bar{c}d) \vee (\bar{a}b\bar{c}d) \vee (a\bar{b}\bar{c}\bar{d}) \vee (a\bar{b}\bar{c}d) \vee (a\bar{b}c\bar{d}) \vee (a\bar{b}cd)$
 Упростим функцию. По ТИ видно, что в наборах 6-7, 10-11 и 14-15 можно попарно сократить переменную d , а в наборах 5-13 сократить переменную a .
 Запишем новое выражение:

$$F = (\bar{a}\bar{b}\bar{c}\bar{d}) \vee (b\bar{c}\bar{d}) \vee (\bar{a}b\bar{c}) \vee (a\bar{b}\bar{c}) \vee (a\bar{b}c).$$



N	a	b	c	d	F
0	0	0	0	0	1
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	0
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	1
7	0	1	1	1	1
8	1	0	0	0	0
9	1	0	0	1	0
10	1	0	1	0	1
11	1	0	1	1	1
12	1	1	0	0	0
13	1	1	0	1	1
14	1	1	1	0	1
15	1	1	1	1	1

В полученном выражении можно минимизировать выражения в 4 и 5 скобках: $(a\bar{b}lc) \vee (a\bar{b}lc) = (alc)$. Далее минимизировать нечего. Итоговое выражение в базисе И, ИЛИ, НЕ: $F = (\bar{a}\bar{b}\bar{c}\bar{d}) \vee (b\bar{c}ld) \vee (\bar{a}b\bar{c}) \vee (alc)$.

4. Кирилл устроился работать помощником системного администратора в IT-отдел Университета. В первый же день работы Кирилл получил ответственное задание. Отдел информационных технологий Университета использует диапазон IP-адресов 192.168.76.0/22, в котором настроено адресное пространство сетей филиалов из 6 подсетей. Данные по распределению адресного пространства частично известны. При этом адресное пространство использовано максимально эффективно.

Кирилл обладает следующими знаниями:

- одному устройству должен соответствовать один IP-адрес вида XXX.XXX.XXX.XXX, при этом XXX – число в диапазоне [0:255], называемое октетом;
- устройства внутри одной подсети должны беспрепятственно взаимодействовать друг с другом и должны группироваться по их расположению;
- выделенный размер подсети содержит минимально необходимое количество адресов;
- подсети большего размера должны находиться в начале адресного диапазона сети;
- в каждом сегменте сети первый адрес зарезервирован под идентификатор подсети, а последний – под широковещательную рассылку;
- выделенный размер подсети записывается как 2^{n-2} , где n – минимальное количество бит, которое может содержать все номера адресов подсети.



Название подсети	Адрес сети	Выделенный размер	Префикс	Десятичная маска	Диапазон доступных адресов	Широковещательный адрес
Главный корпус		510				
Инженерный корпус					192.168.78.1 - ...	
Учебный центр №2	192.168.79.0		/25			
Библиотека						192.168.79.191
Пост охраны				255.255.255.224		
IT-отдел					... - 192.168.79.254	

Заполнение информации вызвало у помощника системного администратора затруднения. Необходимо помочь Кириллу заполнить таблицу недостающими данными. **(20 баллов)**

Решение:

Название подсети	Адрес сети	Выделенный размер	Префикс	Десятичная маска	Диапазон доступных адресов	Широковещательный адрес
Главный корпус	192.168.76.0	510	/23	255.255.254.0	192.168.76.1 - 192.168.77.254	192.168.77.255
Инженерный корпус	192.168.78.0	254	/24	255.255.255.0	192.168.78.1 - 192.168.78.254	192.168.78.255
Учебный центр №2	192.168.79.0	126	/25	255.255.255.128	192.168.79.1 - 192.168.79.126	192.168.79.127
Библиотека	192.168.79.128	62	/26	255.255.255.192	192.168.79.129 - 192.168.79.190	192.168.79.191
Пост охраны	192.168.79.192	30	/27	255.255.255.224	192.168.79.193 - 192.168.79.222	192.168.79.223
IT-отдел	192.168.79.224	30	/27	255.255.255.224	192.168.79.225 - 192.168.79.254	192.168.79.255

5. Отдел закупок шахты «Первая марсианская» получил список доступных запчастей для горно-шахтных погрузчиков, имеющих в наличии на базе снабжения «Лунная 4». В отделе уже был список потребных запчастей от главного инженера шахты. Начальник отдела закупок попросил старшего программиста выяснить, каких из потребных запчастей НЕТ в наличии на базе снабжения, и прислать ему такой список по квантовой почте в виде упорядоченной последовательности артикулов запчастей. Каждый артикул записан в списках заглавными буквами английского алфавита и цифрой от 1 до 3. Помогите старшему программисту шахты составить алгоритм и программу для решения этой задачи, считая, что исходные списки пригодны для непосредственной обработки выбранным языком программирования и уже загружены в память. Правильность решения докажете на простом примере. В решении **нельзя** использовать встроенные в язык операции, методы или свойства для работы со структурами данных, кроме как для определения длины структуры или добавления (удаления) элементов структуры. Код необходимо снабдить комментариями. **(25 баллов)**



Решение:

Алгоритм решения задачи:

Задача на разность множеств. Разность двух упорядоченных множеств **A** и **B** – это множество с элементами **A**, не совпадающими с элементами **B**, без дублей. Сложность алгоритма $O(m+n)$, где **m** и **n** – длины входных упорядоченных множеств **A** и **B** соответственно.

Необходимо написать программу, аргументы которой - два упорядоченных массива для вычисления разности. Функция возвращает третий массив, содержащий элементы, которые есть только в первом массиве и нет во втором. Необходимо взять два исходных массива и пустой третий. Сделать и обнулить три счетчика индексов массивов: $i=0, j=0, k=0$. Сравниваем верхние (нулевые) элементы массивов. Если $M_{1i}=M_{2j}$, то: $i=i+1, j=j+1$ – поднимаем M_1 и M_2 вверх на один элемент. Если $M_{1i}<M_{2j}$, то: $M_{3k}=M_{1i}, i=i+1, k=k+1$ – поднимаем M_1 и M_3 вверх на один элемент. Иначе: $j=j+1$ – поднимаем M_2 вверх на один элемент. И так в цикле while пока $i<L1$ и $j<L2$, где L – длина массива. Если $\max(M_1)<\max(M_2)$, то на этом всё, все элементы перебрали. Если $\max(M_1)>\max(M_2)$, то не перебранными останутся элементы первого массива, превосходящие по величине любой элемент второго массива. Эти элементы включаем в третий массив (последний while в mathcad). Проверить индексацию массивов и правильность определения верхней границы массива в условии цикла while.

Пример решения на PascalABC.NET:

```
1 program Zadacha_5; //Необязательный заголовок
2 //Задача на разность множеств
3 //В языке есть встроенные множества "set"
4 //и метод a.эксерт(b) разности множеств
5 //По условиям задачи так делать нельзя.
6
7 //Динамические массивы с данными
8 var
9 a: array of string := ('AA2', 'BB', 'CC', 'DD1', 'EE');
10 b: array of string := ('AA2', 'DD1', 'DE', 'ED', 'FF');
11 //Выходной массив. изначально пустой
12 c: array of string;
13 i, j, k: integer; //Счётчики=0 по умолчанию
14
15 begin
16 while (i < a.Length) and (j < b.Length) do
17 if a[i] = b[j] then //Ветка true
18 begin
19 i := i + 1;
20 j := j + 1;
21 end
22 else //Ветка false
23 if a[i] < b[j] then
24 begin
25 SetLength(c, k + 1);
26 c[k] := a[i];
27 i := i + 1; k := k + 1;
28 end
29 else j := j + 1;
30 while (i < a.Length) do
31 begin
32 SetLength(c, k + 1);
33 c[k] := a[i];
34 i := i + 1; k := k + 1;
35 end;
36 writeln(c);
37
38 //А теперь с помощью встроенного метода
39 a.Эксерт(b).Println;
40 end.
```

Окно вывода
[BB, CC, EE]
BB CC EE

← Ответ



6. Даша и Арина решили испечь пиццу. Для этого они раскатали на кухонном столе тесто ровно по краям стола. Стол прямоугольной формы, размер стола 120x72 см. Какой диаметр теста для круглой пиццы им нужно сделать, чтобы все пиццы были одинаковыми и максимально большими? Сколько пицц они могут сделать из этого теста? Напишите алгоритм и универсальную программу, которая на основе размеров стола вычисляет и выводит в качестве ответа количество одинаковых круглых заготовок максимального размера из теста, раскатанного по всему столу, и диаметр заготовки. Правильность решения докажите поэтапно вручную по коду. Код необходимо снабдить комментариями. **(25 баллов)**

Решение:

Алгоритм решения задачи:

Задача на использование рекурсии и алгоритма «Разделяй и властвуй». Есть некоторая поверхность прямоугольной формы с заданными сторонами. Ее необходимо разбить на минимальное количество одинаковых квадратов. Соответственно площадь квадратов должна быть максимальной. Решается с помощью «Алгоритма Евклида». Суть алгоритма: определить базовый случай и применить рекурсию. Базовый случай: соотношение сторон участка 2:1. Графическая иллюстрация см. вариант 1. Прямоугольник необходимо разбить на минимальное число квадратов. Ответ: диаметр заготовки = сторона квадрата = 24 см. Проверка поэтапно на рисунке с кодом. Проверка по сторонам: $120:24=5$; $72:24=3$; $5 \times 3=15$ пицц.

Пример решения на PascalABC.NET:

```
•Гранит_2021
1 program Zadacha_6; //Необязательный заголовок
2 //Задача на определение максимального квадрата,
3 //которым можно заполнить прямоугольник.
4 var //Переменные для хранения сторон прямоугольника
5 //a,b - стороны участка, m - остаток от деления mod
6   a1,b1,L: integer;
7 //рекурсивная функция
8 function kvadrat(a,b: integer):integer;
9 var m: integer;
10 begin //если указали стороны в неправильном порядке
11   println('a=',a); //для контроля хода решения
12   println('b=',b);
13   //базовый случай, отношение сторон 2/1 для
14   //подобранных примеров
15   if (a mod b)=0 then //это и есть базовый случай
16     begin
17       Result:=b; //возвращаемое значение
18       Println('Сторона квадрата = ', b);
19     end
20   else //базовый случай не достигнут
21     begin
22       m:=a mod b;
23       Result:=kvadrat(b,m); //запускаем рекурсию
24     end;
25 end;
26
27 begin //основная программа
28   a1:=120; b1:=72;
29   L:=kvadrat(a1,b1); //вызов рекурсивной функции
30   Println('Ответ = ', (a1/L)*(b1/L));
31 end.
```

Окно вывода
a= 120
b= 72
a= 72
b= 48
a= 48
b= 24
Сторона квадрата = 24
Ответ = 15