

# I. Задания заключительного этапа олимпиады 2014-15 года

## Заключительный этап 11 класса (приведен один из вариантов заданий)

### 1. Системы счисления (2 балла)

#### [Последние цифры]

##### Условие

Число, записанное в четверичной системе счисления как  $323_4$ , возвели в степень, записанную в четверичной системе счисления как  $232_4$ . Определите и запишите в ответ подряд без пробелов последние две цифры результата возведения в степень, записанного в четверичной системе счисления.

##### Ответ: 21

##### Решение

Возведение в степень натурального числа есть множественное умножение его на само себя. Обратим внимание, что при перемножении двух натуральных чисел последняя цифра результата зависит только от значений последних цифр перемножаемых чисел. В условии задания требуется определить последние две цифры результата возведения в степень четверичного числа, но мы воспользуемся тем, что исходное число в четверичной системе счисления можно перевести в шестнадцатеричную систему счисления и тогда последние две цифры его четверичной записи будут однозначно соответствовать одной последней цифре его шестнадцатеричной записи:  $323_4 = 3B_{16}$ . Теперь рассмотрим первые несколько степеней этого числа (отметим, что при решении задания достаточно было получать только последние цифры):

$$3B_{16}^1 = 3B_{16}$$

$$3B_{16}^2 = 3B_{16} * 3B_{16} = D99_{16}$$

$$3B_{16}^3 = D99_{16} * 3B_{16} = 32243_{16}$$

$$3B_{16}^4 = 32243_{16} * 3B_{16} = B8E571_{16}$$

$$3B_{16}^5 = B8E571_{16} * 3B_{16} = 2A9CE10B_{16}$$

Обратим внимание, что после возведения в пятую степень последняя цифра оказалась опять **B**, как уже было при возведении в первую степень. Поскольку умножаем мы на одно и то же число, а последняя цифра зависит только от последних цифр сомножителей, можно сделать вывод, что последние цифры будут образовывать цикл из четырех значений: **B-9-3-1**. Значит для того, чтобы определить, какая цифра будет в результате возведения числа  $3B_{16}$  в степень  $232_4 = 46_{10}$  можно посчитать остаток от деления 46 на 4. Остаток равен 2, следовательно, последняя цифра в записи числа в шестнадцатеричной системе счисления будет второй цифрой цикла, то есть 9. Осталось перевести  $9_{16}$  в четверичную систему счисления и получить  $21_4$ . Значит в конце четверичной записи результата возведения в степень будет последовательность цифр 21.

### 2. Кодирование информации. Объем информации (1 балл)

#### [Складской учет]

##### Условие

При поступлении товара на склад необходимо каждой единице товара присвоить уникальный код размещения. Товары размещаются в паллеты так, что в каждой паллете ровно 18 единиц товара. Паллеты размещаются на полках так, что на каждой полке размещается по 32 паллеты. Полки объединены в секции по 256 полок в каждой и всего на складе **X** секций.

Петя разработал программу складского учета, при использовании которой решил использовать следующую методику формирования и хранения уникального кода размещения товара. Уникальный код размещения единицы товара представлял собой **одно десятичное число**, состоящее из четырех групп разрядов. В старшей группе разрядов записывался номер секции, в следующей за ней группе разрядов записывался номер полки в секции, в следующей за ней группе разрядов записывался номер паллеты на полке и в младшей группе разрядов записывался порядковый номер единицы товара в паллете. Количество разрядов в каждой группе таково, чтобы была возможность записать максимальный номер соответствующего объекта хранения (секции, полки, паллеты или единицы товара в паллете). Далее Петя определил максимальное число **Y**, которое у него могло получиться в качестве уникального кода размещения товара по его методике, и решил выделить для хранения каждого уникального кода размещения единицы товара одинаковое минимальное количество бит памяти, необходимое для хранения любых чисел, больших **0** и не превосходящих **Y**.

Вася увидел нерациональность такого подхода и придумал собственную методику формирования и хранения уникального кода размещения товара. Он решил каждой комбинации из возможных номеров секций, полок в секции, паллет на полке и товаров в паллете поставить в соответствие натуральное число от **1** до **Z**, так, что каждое число из этого диапазона соответствует одной и только комбинации. Такое число и является уникальным кодом размещения товара в методике Васи, а для хранения этих кодов Вася решил использовать одинаковое минимально возможное количество бит памяти. В результате Вася обнаружил, что его методика привела к уменьшению объема памяти, необходимого для хранения одного уникального кода размещения товара ровно на **7** бит. Определите минимальное количество секций на складе **X**, при котором это возможно. В ответе запишите целое число.

##### Ответ: 7

##### Решение

Максимальное число **Y** для методики Пети можно записать как  $X2563218$ , где вместо **X** необходимо подставить максимальный номер секции. Тогда для хранения кода размещения единицы товара по методике Пети необходимо **A** бит, где **A** можно посчитать как  $\log_2(X * 10000000 + 2563218)$  с округлением в большую сторону до ближайшего целого числа. Число комбинаций для методики Васи будет равно  $X * 256 * 32 * 18 = X * 147456$ . Тогда для хранения кода размещения единицы товара по методике Васи необходимо **B** бит, где **B** можно посчитать как  $\log_2(X * 147456)$  с округлением в большую сторону до ближайшего целого числа. Следовательно, необходимо найти минимальное число **X**, такое, что  $A - B = 7$ . Непосредственно поиск такого числа **X** можно осуществлять по-разному. С учетом больших числовых значений, можно, например, воспользоваться электронными таблицами:

|    | A  | B  | C                                | D      |
|----|----|--|----------------------------------|--------|
| 1  | 1  | =ОКРУГЛВВЕРХ(LOG(A1*10000000+2563218;2);0) | =ОКРУГЛВВЕРХ(LOG(A1*147456;2);0) | =B1-C1 |
| 2  | 2  |  |                                  |        |
| 3  | 3  |  |                                  |        |
| 4  | 4  |  |                                  |        |
| 5  | 5  |  |                                  |        |
| 6  | 6  |  |                                  |        |
| 7  | 7  |  |                                  |        |
| 8  | 8  |  |                                  |        |
| 9  | 9  |  |                                  |        |
| 10 | 10 |  |                                  |        |

Копируя ячейки с формулами в столбцах B, C и D в нижележащие ячейки этих же столбцов легко убедиться, что первый раз число 7 появиться в седьмой строке, что и даст правильный ответ задания.

### 3. Основы логики (3 балла)

#### [Логическое тождество]

##### Условие

Определите, сколько существует неэквивалентных друг другу логических функций трех переменных  $F(X,Y,Z)$ , таких, что выполняется тождественное равенство:

$$((F \rightarrow X) \rightarrow Y) \rightarrow Z = ((X \rightarrow Y) \text{ and } \text{not } Z) \rightarrow \text{not } F$$

В ответе укажите целое число.

Примечание. Две логические функции трех переменных  $F(X,Y,Z)$  и  $G(X,Y,Z)$  будем считать неэквивалентными друг другу тогда и только тогда, когда существует хотя бы один набор значений переменных  $(X',Y',Z')$ , такой, что значение функции  $F(X',Y',Z')$  не равно значению функции  $G(X',Y',Z')$ .

**Ответ: 64**

##### Решение

Для того чтобы задать логическую функцию трех переменных необходимо определить ее значения на восьми возможных комбинациях значений переменных  $X, Y$  и  $Z$ . Отметим, что всего существует 256 неэквивалентных друг другу логических функций трех переменных. Условие задачи накладывает на  $F(X,Y,Z)$  единственное ограничение – чтобы сохранялось тождество, левая и правая часть которого содержит только  $F(X,Y,Z)$  и переменные  $X, Y$  и  $Z$ . Таким образом, нам необходимо проверить накладывает ли тождество ограничение на значение функции, соответствующее каждой комбинации значений ее параметров. Для этого, вычислим левую и правую часть тождества для каждой комбинации, подставляя соответствующие значения  $X, Y$  и  $Z$ :

| Комбинация значений параметров | $((F \rightarrow X) \rightarrow Y) \rightarrow Z$ | $((X \rightarrow Y) \text{ and } \text{not } Z) \rightarrow \text{not } F$ |
|--------------------------------|---|--|
| 0 0 0                          | not F   | not F  |
| 0 0 1                          | 1   | 1  |
| 0 1 0                          | 0   | not F  |
| 0 1 1                          | 1   | 1  |
| 1 0 0                          | 1   | 1  |
| 1 0 1                          | 1   | 1  |
| 1 1 0                          | 0   | not F  |
| 1 1 1                          | 1   | 1  |

Проанализируем получившуюся таблицу. Обратим внимание, что только две строки, соответствующие комбинациям значений параметров (0 1 0) и (1 1 0) накладывают ограничения на значения функции  $F(X,Y,Z)$ : функция должна быть истинна при этих комбинациях значений параметров, для того, чтобы выполнялось тождество. При всех остальных комбинациях параметров тождество будет истинным независимо от значений функции  $F(X,Y,Z)$ . Следовательно, таблица истинности функции  $F(X,Y,Z)$  будет выглядеть следующим образом:

| Комбинация значений параметров | $F(X,Y,Z)$ |
|--------------------------------|------------|
| 0 0 0                          | ?          |
| 0 0 1                          | ?          |
| 0 1 0                          | 1          |
| 0 1 1                          | ?          |
| 1 0 0                          | ?          |
| 1 0 1                          | ?          |
| 1 1 0                          | 1          |
| 1 1 1                          | ?          |

Вопросительным знаком обозначены значения функции, которые могут быть равны как 0, так и 1 не нарушая тождества. Таких значений 6. Следовательно, существует  $2^6=64$  логических функции трех переменных, удовлетворяющих условию задачи.

#### 4. Кодирование информации. Алгоритмы обработки кодированной информации (1 балл) [Все подряд]

##### Условие

Последовательность из 0 и 1 построена следующим образом. Взяты все натуральные числа от 1 до 65536. Каждое число переведено в двоичную систему счисления. Получившиеся двоичные записи чисел (только значащие цифры) помещены друг за другом в порядке возрастания соответствующих натуральных чисел. Вот начало этой последовательности:

11011100101110111100010011010...

Определите, где в этой последовательности **первый раз** встретятся идущие подряд 9 единиц. В ответе укажите одно целое число – номер позиции в последовательности, считая от 1 слева направо, начиная с которой записаны эти 9 единиц.

**Ответ: 1786**

##### Решение

Проанализируем начало последовательности. Разными цветами выделены участки, соответствующие одноразрядным, двухразрядным, трехразрядным двоичным кодам и т.д. (для наглядности продолжим последовательность до появления пятиразрядного кода):

110111001011101111000100110101011110011011110111110000...

Заметим, что последовательность из первых трех идущих подряд единиц образуется последним двухразрядным кодом (11), который состоит из одних единиц и первым трехразрядным кодом (100), который начинается с одной единицы. Аналогично последовательность из первых четырех идущих подряд единиц образуется последним трехразрядным кодом (111), который состоит из одних единиц и первым четырехразрядным кодом (1000), который начинается с одной единицы. Легко заметить, что встретить последовательность из  $n$  единиц ранее, чем на стыке с началом участка  $n$ -разрядных кодов невозможно. Определим, где будет начало участка 9-ти разрядных последовательностей. Для этого необходимо сложить длины участков последовательностей с меньшим количеством разрядов. Заметим, что каждый участок с  $n$ -разрядными кодами будет иметь длину  $n \cdot 2^{n-1}$ . Это обусловлено тем, что всего существует  $2^n$  различных двоичных кодов длиной  $n$  символов, но мы берем только половину от них – те, которые начинаются с единицы, то есть  $2^{n-1}$ . Таким образом, коды, имеющие количество разрядов меньше 9 будут образовывать последовательность, длиной:

$$1 \cdot 2^0 + 2 \cdot 2^1 + 3 \cdot 2^2 + 4 \cdot 2^3 + 5 \cdot 2^4 + 6 \cdot 2^5 + 7 \cdot 2^6 + 8 \cdot 2^7 = 1793$$

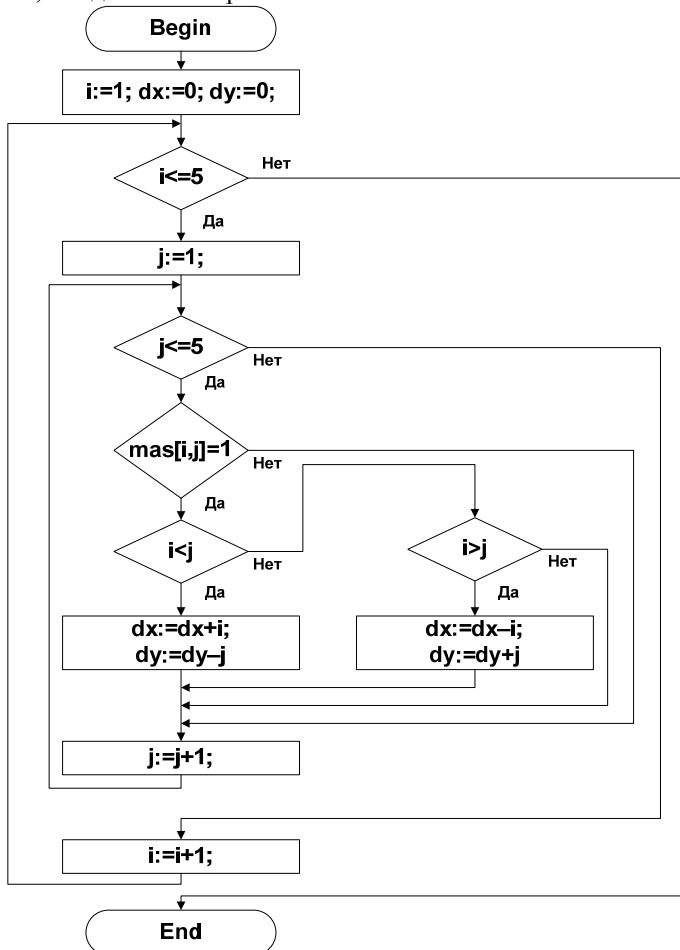
Последние 8 символов этой последовательности будут первыми восемью единицами в искомой девятиразрядной последовательности единиц. Следовательно, первый символ этой последовательности будет иметь номер  $1793 - 8 + 1 = 1786$ .

#### 5. Алгоритмизация и программирование. Анализ алгоритма, заданного в виде блок-схемы (3 балла)

##### [Матрица перемещения]

##### Условие

Дана блок-схема алгоритма. На вход алгоритма можно подать только двумерный массив **mas**, размером 5 на 5 элементов, каждый из которых может быть 0 или 1.



Сколько существует различных двумерных массивов, которые можно подать на вход алгоритма таких, что после его завершения получатся значения  $dx=-1$  и  $dy=-1$ ? В ответе укажите целое число.

Примечания. Нумерация обоих индексов двумерного массива начинается с 1. Двумерные массивы будем считать различными, если они различаются значением хотя бы одного своего элемента.

**Ответ: 128**

**Решение**

Проанализируем блок-схему и отметим следующие особенности алгоритма:

1. Элементы, которые находятся на главной диагонали, то есть элементы, у которых  $i=j$  не могут привести к изменению значения переменных  $dx$  и  $dy$ .
2. Каждый равный 1 элемент не находящийся на главной диагонали меняет значение переменных  $dx$  и  $dy$ , причем это изменение различно, в зависимости от того, находится ли такой единичный элемент выше или ниже главной диагонали. При этом заметим, что любая единица, встреченная выше главной диагонали, приведет к увеличению значения переменной  $dx$  на величину меньшую, чем та на которую уменьшится значение переменной  $dy$ . Аналогично, любая единица, встреченная ниже главной диагонали, приведет к увеличению значения переменной  $dy$  на величину меньшую, чем та на которую уменьшится значение переменной  $dx$ .

Тогда существует только 4 комбинации значений элементов, не лежащих на главной диагонали, которые приводят к тому, что значения переменных  $dx$  и  $dy$  становятся одновременно равны  $-1$  после завершения алгоритма. Это комбинации, содержащие ровно две единицы, причем расположенные симметрично непосредственно над и под главной диагональю:

|   |   |   |   |
|---|---|---|---|
| $\begin{bmatrix} ? & 1 & 0 & 0 & 0 \\ 1 & ? & 0 & 0 & 0 \\ 0 & 0 & ? & 0 & 0 \\ 0 & 0 & 0 & ? & 0 \\ 0 & 0 & 0 & 0 & ? \end{bmatrix}$ | $\begin{bmatrix} ? & 0 & 0 & 0 & 0 \\ 0 & ? & 1 & 0 & 0 \\ 0 & 1 & ? & 0 & 0 \\ 0 & 0 & 0 & ? & 0 \\ 0 & 0 & 0 & 0 & ? \end{bmatrix}$ | $\begin{bmatrix} ? & 0 & 0 & 0 & 0 \\ 0 & ? & 0 & 0 & 0 \\ 0 & 0 & ? & 1 & 0 \\ 0 & 0 & 1 & ? & 0 \\ 0 & 0 & 0 & 0 & ? \end{bmatrix}$ | $\begin{bmatrix} ? & 0 & 0 & 0 & 0 \\ 0 & ? & 0 & 0 & 0 \\ 0 & 0 & ? & 0 & 0 \\ 0 & 0 & 0 & ? & 1 \\ 0 & 0 & 0 & 1 & ? \end{bmatrix}$ |
|---|---|---|---|

Знаки вопроса на главной диагонали означают, что значения этих элементов могут быть любыми, так как они не влияют на изменения значений целевых переменных. Следовательно, для каждой из приведенных выше комбинаций значений элементов вне главной диагонали существует  $2^5$  комбинаций значений элементов главной диагонали. Значит, общее количество различных массивов, удовлетворяющих условию, будет  $4 \cdot 2^5 = 128$ .

## 6. Алгоритмизация и программирование. Формальные исполнители (2 балла)

**[Робот-кладоискатель]**

**Условие**

Робот-кладоискатель перемещается по квадратному клетчатому полю, размером 6 на 6 клеток.

Часть клеток поля содержит клады монет. Числа в клетках указывают, что в соответствующей клетке есть клад из этого количества монет.

|          |           |          |          |          |          |
|----------|-----------|----------|----------|----------|----------|
|          | <b>10</b> |          | <b>1</b> |          | <b>В</b> |
| <b>3</b> |           | <b>4</b> |          | <b>1</b> |          |
| <b>7</b> |           |          | <b>4</b> |          |          |
|          |           | <b>3</b> |          |          | <b>5</b> |
|          | <b>6</b>  |          | <b>2</b> |          |          |
| <b>А</b> |           | <b>3</b> |          | <b>9</b> | <b>6</b> |

Робот ищет клад, руководствуясь следующими правилами:

1. За один ход робот может перемещаться на одну клетку вправо, влево, вверх или вниз, не выходя за пределы поля.
2. На каждый ход робот тратит одну единицу энергии.
3. Если робот попадает в клетку с кладом, он забирает указанное в ней количество монет.
4. Батарейка робота содержит ровно 16 единиц энергии.
5. Перед началом движения робот находится в клетке «А». В ней нет клада.
6. Если робот попадает в клетку «В», он завершает движение, даже если у него осталась энергия. В клетке «В» нет клада.
7. Миссия робота считается успешной, только если он попал в клетку «В».

Определите, какое максимальное количество монет может быть у робота в результате успешного завершения миссии. В ответе укажите целое число.

**Ответ: 36**

**Решение**

Анализировать возможные пути робота вручную при заданных условиях представляется, во-первых, трудоемким процессом, а во-вторых, не гарантирует, что найденный путь действительно будет позволять собрать максимальное

количество монет. Поэтому одним из вариантов решения задания является программная реализация алгоритма, перебирающего возможные траектории движения робота. Пример реализации рекурсивного алгоритма приведен ниже:

```
Type
  arr = array[1..6, 1..6] of integer;
const
  map: arr =
    ((0,10,0,1,0,0),
     (3,0,4,0,1,0),
     (7,0,0,4,0,0),
     (0,0,3,0,0,5),
     (0,6,0,2,0,0),
     (0,0,3,0,9,6));
  fuel : integer = 16;
var
  prize, max_prize, tr_length : integer;
  current_map : arr;
  trajectory : array[1..2,1..16] of integer; //1=X, 2=Y

procedure move(X:integer; Y:integer);
var
  i : integer;
begin
  inc (tr_length);
  trajectory[1,tr_length]:=X;
  trajectory[2,tr_length]:=Y;
  if (X=1) and (Y=6) then
    begin
      prize:=0;
      current_map:=map;
      for i := 1 to tr_length do
        begin
          prize:=prize+current_map[trajectory[1,i],trajectory[2,i]];
          current_map[trajectory[1,i],trajectory[2,i]]:=0;
        end;
      if prize>=max_prize then max_prize:=prize;
      dec(tr_length);
      exit;
    end;
  if tr_length=fuel then
    begin
      dec(tr_length);
      exit;
    end;
  if Y<6 then move(X,Y+1);
  if X<6 then move(X+1,Y);
  if X>1 then move(X-1,Y);
  if Y>1 then move(X,Y-1);
  dec(tr_length);
end;

begin
  max_prize:=0;
  tr_length:=0;
  move(5,1);
  tr_length:=0;
  move(6,2);
  writeln(max_prize);
end.
```

Выполнив программу получаем, что максимальное количество монет, которое можно собрать равняется 36. Существует несколько траекторий, которые могут привести к сбору этого количества монет, например:

5,1 | 4,1 | 3,1 | 4,1 | 5,1 | 5,2 | 6,2 | 6,3 | 6,4 | 6,5 | 6,6 | 5,6 | 4,6 | 3,6 | 2,6 | 1,6

Указаны адреса клеток, которые посещает робот, собирая монеты.

## 7. Телекоммуникационные технологии (2 балла).

### [NAT]

#### Условие

Студенты Иван и Петр изучают сетевые технологии и в частности, технологию трансляции адресов (NAT – Network Address Translation) стека TCP/IP. Эта технология обеспечивает доступ с множества компьютеров из одной сети в другую (внешнюю) сеть через шлюз, который имеет IP-адрес как в первой сети, так и во внешней сети.

При взаимодействии через стек TCP/IP компьютеры идентифицируются по IP-адресу - адресу протокола сетевого уровня.

При взаимодействии через стек TCP/IP приложения на компьютере идентифицируются по номеру порта протокола транспортного уровня (у каждого запущенного на одном компьютере приложения номер порта уникальный). Программы могут быть программами-серверами, и тогда номер порта по умолчанию известен заранее для каждого прикладного протокола, или программами-клиентами, которые занимают свободный порт. На компьютере может быть запущено несколько программ-клиентов, и работать они будут с портов с разными номерами.

Устанавливает сеанс работы между приложением-сервером и приложением-клиентом, как раз программа-клиент. И сеанс идентифицируется адресом компьютера клиента, номером порта программы-клиента, адресом компьютера сервера, номером порта сервера.

Принцип работы NAT следующий: сообщения от компьютера-клиента к компьютеру-серверу, расположенному во внешней сети, попадают в шлюз, где в сообщении адрес отправителя (компьютера-клиента) и порт программы-клиента подменяются на внешний адрес шлюза и свободный порт шлюза. Таким образом, запрос к компьютеру-серверу идет как бы от самого шлюза. Запись о замене (запись трансляции) заносится в специальную таблицу и используется при обратной замене адреса и порта на исходные при поступлении ответа от внешнего компьютера. За счет использования разных внешних портов можно устанавливать связи с одной программой-сервером множеству программ-клиентов с разных локальных компьютеров.

Иван и Петр построили в среде моделирования сеть с NAT. Модель сети включает:

1. Внешнюю сеть с 2 компьютерами-серверами с адресами 10.0.0.1 и 10.0.0.2, на каждом из которых запущена программа-сервер S-File, которая работает по порту с номером 456.
2. Внутреннюю сеть с двумя компьютерами – одним для Петра и одним для Ивана с адресами 192.168.0.1 и 192.168.0.2 соответственно. На этих компьютерах студенты могут запускать программы-клиенты C-File по одному экземпляру для скачивания одного файла. Каждый экземпляр C-File занимает отдельный порт. При скачивании каждого файла запускается программа-клиент C-File и пытается установить сеанс с соответствующим сервером. Если сеанс удалось установить, программа-клиент незамедлительно начинает скачивать файл. Как только файл скачан целиком, программа-клиент завершает сеанс и завершает свою работу.
3. Шлюз с транслятором адресов, с таблицей трансляции, которая может в один момент времени хранить записи не более чем о 6 сеансах. При установлении сеанса, когда в таблице есть свободные места, в нее добавляется запись о трансляции. По завершении сеанса, относящаяся к нему запись трансляции мгновенно удаляется. Если таблица полна, то приложению, запросившему связь через шлюз, отправляется сообщение о невозможности установить соединение в текущий момент. Внутренний интерфейс шлюза имеет адрес 192.168.0.100, внешний – 10.0.0.100.

В модели передача данных по сети между программами-серверами и программами-клиентами осуществляется с постоянной скоростью 10 КБайт в секунду независимо от количества одновременных потоков передачи данных. Установление сеанса происходит мгновенно.

Используемые протоколы прикладного, транспортного и сетевого уровней совместно добавляют к каждому КБайту данных 256 байт служебных данных.

Если программа C-File не может установить соединение с программой S-File, она автоматически повторяет попытку через 5 секунд.

Быстродействие сетевых интерфейсов компьютеров и шлюза, размер и скорость работы их буферов, и скорость линий связи в модели позволяют не учитывать их влияние при оценке времени передачи данных.

Также при расчетах следует пренебречь вкладом всех остальных (кроме описанных) процедур сетевого взаимодействия в общий объем переданных данных и всеми другими (кроме описанных) задержками на устройствах и программном обеспечении.

Время в модели считается кратно 1 секунде.

Через какое время в секундах от начала работы модели Иван и Петр закончат скачивать файлы на свои компьютеры при условии что:

1. Одновременно с началом работы модели Иван начал скачивать файл f1 объемом 112 КБайт с сервера 10.0.0.1 .
2. Спустя 4 секунды после начала работы Петр начал скачивать файл f2 объемом 128 КБайт с сервера 10.0.0.2 .
3. Спустя 8 секунд после начала работы Иван начал скачивать файл f3 объемом 64 КБайт с сервера 10.0.0.1 .
4. Спустя 6 секунд после начала работы Петр начал скачивать файл f4 объемом 64 КБайт с сервера 10.0.0.2 .
5. Спустя 9 секунд после начала работы Петр начал скачивать файл f5 объемом 48 КБайт с сервера 10.0.0.1 .
6. Спустя 8 секунд после начала работы Петр начал скачивать файл f6 объемом 32 КБайт с сервера 10.0.0.1 .
7. Спустя 10 секунд после начала работы Иван начал скачивать файл f7 объемом 96 КБайт с сервера 10.0.0.2 .
8. Спустя 12 секунд после начала работы Иван начал скачивать файл f8 объемом 80 КБайт с сервера 10.0.0.1 .

В ответе укажите целое число.

**Ответ: 27**

**Решение**

Проанализируем условие задачи и заметим, что если удастся установить соединение, то скачивание какого-либо файла осуществляется с одной и той же постоянной скоростью и, соответственно, время его скачивания будет определяться только объемом скачиваемых данных. Единственное ограничение, которое может повлиять на время завершения скачивания файла – это размер таблицы трансляции, равный 6, то есть одновременно может реализовываться не более 6 потоков передачи данных. Построим модель, демонстрирующую описанные в условии процессы передачи данных:

| Имя файла                                | Размер файла, КБайт | Объем передаваемых данных, КБайт | Время, требуемое на передачу данных, с | Момент первой попытки установить соединение, с | Время, с |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |
|--|---------------------|----------------------------------|--|--|----------|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|
|  |                     |                                  |  |  | 1        | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 |   |   |   |   |
| f1                                       | 112                 | 140                              | 14                                     | 0  | 1        | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1  | 1  | 1  | 1  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |
| f2                                       | 128                 | 160                              | 16                                     | 5  |          |   |   |   | 1 | 1 | 1 | 1 | 1 | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  |    |    |    |    |    |    |    |    |    |   |   |   |   |
| f3                                       | 64                  | 80                               | 8                                      | 9  |          |   |   |   |   |   |   |   | 1 | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  |    |    |    |    |    |    |    |    |    |    |   |   |   |   |
| f4                                       | 64                  | 80                               | 8                                      | 7  |          |   |   |   |   | 1 | 1 | 1 | 1 | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  |    |    |    |    |    |    |    |    |    |    |   |   |   |   |
| f5                                       | 48                  | 60                               | 6                                      | 10   |          |   |   |   |   |   |   |   |   | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  |    |    |    |    |    |    |    |    |    |    |   |   |   |   |
| f6                                       | 32                  | 40                               | 4                                      | 9  |          |   |   |   |   |   |   |   | 1 | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  |    |    |    |    |    |    |    |    |    |    |   |   |   |   |
| f7                                       | 96                  | 120                              | 12                                     | 11   |          |   |   |   |   |   |   |   |   | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  |    |    |   |   |   |   |
| f8                                       | 80                  | 100                              | 10                                     | 13   |          |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  |    |    |   |   |   |   |
| Количество записей в таблице трансляции: |                     |                                  |  |  | 1        | 1 | 1 | 1 | 1 | 2 | 2 | 3 | 3 | 5  | 6  | 6  | 6  | 6  | 6  | 6  | 6  | 4  | 4  | 3  | 3  | 3  | 3  | 2  | 2  | 1  | 1  | 1 | 1 | 1 | 1 |

Столбцы «Имя файла» и «Размер файла» берутся из условия задания. Столбец «Объем передаваемых данных» вычисляется как «Размер файла \* 1,25», поскольку по условию задания к каждому КБайту передаваемых данных добавляется еще 256 байт служебных. Столбец «Время, требуемое на передачу данных» вычисляется как «Объем передаваемых данных / 10», поскольку по условию задания передача данных происходит с постоянной скоростью 10 КБайт в секунду. В столбце «Момент первой попытки установить соединение» указывается номер секунды, в которую будет происходить первая попытка установить соединение для скачивания соответствующего файла. Эти данные берутся из условия задания. Дальше следуют столбцы, соответствующие секундам передачи данных. Если файл передается в эту секунду, то в соответствующей ячейке указана 1. Если была неуспешная попытка установить соединение – в ячейке указан 0. Ячейка, выделенная красным цветом, показывает секунду, в которую произошла попытка установить соединение для скачивания файла f7, которая оказалась неудачной из-за того, что таблица трансляции в этот момент полностью занята. Следующая попытка установить соединение по условию задания осуществляется через 5 секунд.

Из представленной модели видно, что скачивание последнего файла завершится по истечении 27-ой секунды.

**8. Технологии обработки информации в электронных таблицах (1 балл)**  
**[Окошки]**

**Вариант 1**

Дана электронная таблица в режиме отображения формул:

|    | A | B | C   | D | E | F  |
|----|---|---|---|---|---|----|
| 1  |   |   | 2   | 4 | 8 | 16 |
| 2  |   | 0 | =ОСТАТ(ЧАСТНОЕ(\$A\$1;СТЕПЕНЬ(C\$1;\$B2));C\$1) |   |   |    |
| 3  |   | 1 |   |   |   |    |
| 4  |   | 2 |   |   |   |    |
| 5  |   | 3 |   |   |   |    |
| 6  |   | 4 |   |   |   |    |
| 7  |   | 5 |   |   |   |    |
| 8  |   | 6 |   |   |   |    |
| 9  |   | 7 |   |   |   |    |
| 10 |   | 8 |   |   |   |    |

Ячейку C2 скопировали во все ячейки диапазона C2:F10. Какое минимальное натуральное число необходимо поместить в ячейку A1, для того, чтобы в результате вычислений в незакрашенных ячейках получились значения, как на рисунке:

|    | A | B | C | D | E | F  | G |
|----|---|---|---|---|---|----|---|
| 1  |   |   | 2 | 4 | 8 | 16 |   |
| 2  |   | 0 |   |   |   |    | 1 |
| 3  |   | 1 |   |   | 2 |    |   |
| 4  |   | 2 |   |   |   |    |   |
| 5  |   | 3 |   | 3 |   |    |   |
| 6  |   | 4 |   |   |   |    |   |
| 7  |   | 5 |   |   |   |    |   |
| 8  |   | 6 |   |   |   |    |   |
| 9  |   | 7 |   |   |   |    |   |
| 10 |   | 8 | 1 |   |   |    |   |
| 11 |   |   |   |   |   |    |   |

В ответе укажите целое число.

Примечание. Сопоставление названий функций в Microsoft Excel и OpenOffice.org Calc:

| Microsoft Excel (Rus) | Microsoft Excel (Eng) | OpenOffice.org Calc |
|-----------------------|-----------------------|---------------------|
| ОСТАТ                 | MOD                   | MOD                 |
| СТЕПЕНЬ               | POWER                 | POWER               |
| ЧАСТНОЕ               | QUOTIENT              | QUOTIENT            |

**Ответ: 465**

**Решение**

Обратим внимание, что формулы, в столбцах C, D, E и F вычисляют значения разрядов записи числа, помещенного в ячейку A1 в системах счисления с основаниями 2, 4, 8 и 16 соответственно. Следовательно, нам нужно найти минимальное натуральное число, такое, что его запись в шестнадцатеричной системе счисления будет иметь младшим разрядом 1, в восьмеричной системе счисления вторым справа разрядом – 2, в четверичной системе счисления четвертым справа разрядом 3, а в двоичной системе счисления – девятым справа разрядом – 1. Воспользовавшись фактом, что каждый разряд записи числа в шестнадцатеричной, восьмеричной или четверичной системе счисления эквивалентен четырем, трем или двум разрядам соответственно записи этого числа в двоичной системе счисления, составим таблицу известных значений двоичных разрядов числа:

|                |   |   |   |   |   |   |   |   |   |
|----------------|---|---|---|---|---|---|---|---|---|
| $??I_{16}$     | ? | ? | ? | ? | ? | 0 | 0 | 0 | 1 |
| $??2?_8$       | ? | ? | ? | 0 | 1 | 0 | ? | ? | ? |
| $?3????_4$     | ? | 1 | 1 | ? | ? | ? | ? | ? | ? |
| $1?????????_2$ | 1 | ? | ? | ? | ? | ? | ? | ? | ? |

Из таблицы видно, что нам известны все разряды этого числа и его запись в двоичной системе счисления:

$$111010001_2 = 465_{10}$$

## 9. Технологии сортировки и фильтрации данных (1 балл)

[Фильтрация]

Вариант 1

Петя получил таблицу значений:

| Номер набора | A  | B  | C  |
|--------------|----|----|----|
| Набор 1      | 23 | 23 | 45 |
| Набор 2      | 46 | 47 | 45 |
| Набор 3      | 45 | 44 | 44 |
| Набор 4      | 46 | 22 | 44 |
| Набор 5      | 47 | 23 | 22 |
| Набор 6      | 44 | 45 | 23 |
| Набор 7      | 33 | 46 | 46 |
| Набор 8      | 45 | 44 | 43 |
| Набор 9      | 22 | 44 | 45 |
| Набор 10     | 47 | 45 | 46 |

Он заметил, что если применить к этой таблице фильтр:

**(A<X) ИЛИ (B<X) И (C<X)**

то существует такое целое положительное значение X, при котором удвоение значений в **любом одном** столбце (A, B или C) уменьшает на 3 количество строк удовлетворяющих фильтру. То есть, если удвоить значения в столбце A при сохранении значений в остальных столбцах или удвоить значения в столбце B при сохранении значений в остальных столбцах или удвоить значения в столбце C при сохранении значений в остальных столбцах, в любом из трех случаев



количество строк, удовлетворяющих фильтру, будет на 3 меньше, чем при применении фильтра к исходной таблице. Найдите и вы это число.

В ответе укажите число.

**Ответ: 45**

**Решение**

Дополним исходную таблицу столбцами с удвоенными значениями исходных столбцов  $A$ ,  $B$  и  $C$ .

|          | $A$ | $B$ | $C$ | $2*A$ | $2*B$ | $2*C$ |
|----------|-----|-----|-----|-------|-------|-------|
| Набор 1  | 23  | 23  | 45  | 46    | 46    | 90    |
| Набор 2  | 46  | 47  | 45  | 92    | 94    | 90    |
| Набор 3  | 45  | 44  | 44  | 90    | 88    | 88    |
| Набор 4  | 46  | 22  | 44  | 92    | 44    | 88    |
| Набор 5  | 47  | 23  | 22  | 94    | 46    | 44    |
| Набор 6  | 44  | 45  | 23  | 88    | 90    | 46    |
| Набор 7  | 33  | 46  | 46  | 66    | 92    | 92    |
| Набор 8  | 45  | 44  | 43  | 90    | 88    | 86    |
| Набор 9  | 22  | 44  | 45  | 44    | 88    | 90    |
| Набор 10 | 47  | 45  | 46  | 94    | 90    | 92    |

Отметим, что нет необходимости рассматривать ситуацию, когда  $X < 24$ , поскольку в этом случае до удвоения значения какого-либо столбца в результате применения фильтра не может быть отобрано 3 или более строк.

При любом  $X$ , принадлежащем диапазону значений от 24 до 33 включительно, до удвоения значений какого-либо столбца в результате фильтрации будет отбираться ровно 3 строки (Набор 1, Набор 5, Набор 9), но при этом удвоение значений в столбцах  $B$  или  $C$  будет приводить к тому, что в результате применения фильтра также будут отбираться две строки (Набор 1 и Набор 9). Это происходит потому, что если строки были отобраны по условию  $A < X$  до удвоения значений столбцов  $B$  или  $C$ , то очевидно это условие будет выполняться и после удвоения их значений, а значит строки, которые соответствуют этому условию, будут также отбираться фильтром. Следовательно, разность между количеством отобранных строк не будет равна 3.

При любом  $X$ , принадлежащем диапазону значений от 34 до 44 включительно до удвоения значений какого-либо столбца в результате фильтрации будет отбираться ровно 4 строки (Набор 1, Набор 5, Набор 7, Набор 9), но при этом легко заметить, что три из них будут отбираться за счет выполнения условия  $A < X$ , а значит, как и в предыдущем случае, попадут в результат работы фильтра и после удвоения значений столбца  $B$  или  $C$ . Значит и в этом случае разность количества строк, удовлетворяющих фильтру, не будет равна 3.

А вот при  $X=45$  до удвоения значений какого-либо столбца будет отбираться 8 строк (все строки, кроме Набор 2 и Набор 10), а после удвоения значений в любом столбце будет оставаться ровно по 5 строк, удовлетворяющих фильтру. Таким образом, это значение  $X$  соответствует требованию условия задания. Также отметим, что при значении  $X=46$  количество строк, удовлетворяющих фильтру после удвоения какого-либо столбца, станет равно 7, но до удвоения значения какого-либо столбца фильтру по-прежнему будут удовлетворять только 8 строк. При  $X=47$  и больше после удвоения значений любого столбца не менее 9 строк будут удовлетворять значению фильтра, что делает невозможным получение разницы в количестве строк, равной 3. Следовательно, у задания есть единственный правильный ответ – 45.

## 10. Технологии программирования (2 балла)

Вы разрабатываете систему управления персоналом в некоторой фирме. В фирме работают  $n$  сотрудников. Каждый сотрудник характеризуется своим порядковым номером, который является натуральным числом от одного до  $n$ . Номера всех сотрудников различны.

Сотрудник, имеющий порядковый номер один, является главой фирмы. У каждого из всех остальных сотрудников имеется один непосредственный начальник — сотрудник фирмы, имеющий меньший порядковый номер.

Первая операция, которую вам необходимо реализовать в вашей системе управления персоналом — отправка в командировку одного из сотрудников. Известно, что при отправке в командировку некоторого сотрудника вместе с ним в командировку отправляются все сотрудники, которые подчинены ему непосредственно или через свое начальство.

Вам необходимо, зная структуру фирмы и номер отправляемого в командировку сотрудника, сообщить, сколько сотрудников отправятся в командировку в результате выполнения этой операции.

**Формат входного файла**

В первой строке входного файла `input.txt` находится натуральное число  $n$  ( $2 \leq n \leq 100$ ) — количество сотрудников в фирме.

Следующая строка содержит  $n-1$  натуральное число — номера непосредственных начальников сотрудников с номерами от 2 до  $n$  в соответствующем порядке. Числа отделены друг от друга одним пробелом. Гарантируется, что номер непосредственного начальника очередного сотрудника меньше номера самого сотрудника.

Следующая строка содержит одно натуральное число  $x$  ( $1 \leq x \leq n$ ) — номер отправляемого в командировку сотрудника.

**Формат выходного файла**

В выходной файл `output.txt` выведите одно число — количество сотрудников, отправляющихся в командировку после выполнения описанной операции.

**Пример входных и выходных данных**

| input.txt       | output.txt |
|-----------------|------------|
| 9               | 5          |
| 1 2 1 4 4 2 7 8 |            |
| 2               |            |

## 11. Технологии программирования (4 балла)

Вы разрабатываете систему, одной из задач которой является составление расписания работы над задачами, выполняемыми некоторой организацией, занимающейся разработкой программного обеспечения.

Известно, что организации необходимо выполнить  $n$  задач, пронумерованных натуральными числами от одного до  $n$ . На выполнение каждой задачи требуется ровно один день, и в каждый день может быть выполнена только одна задача. Таким образом, на выполнение всех задач потребуются  $n$  дней, а расписание выполнения задач выглядит как назначение определенного дня на выполнение каждой задачи. Для каждой задачи известно также число  $a_i$  — номер дня, ранее которого не может быть начато выполнение этой задачи.

Ваша программа должна, имея информацию о количестве задач и днях, ранее которых выполнение задач не может быть начато, определить количество допустимых расписаний выполнения задач — количество таких расписаний, при которых каждая задача выполняется не ранее указанного для нее дня.

### Формат входного файла

В первой строке входного файла `input.txt` находится натуральное число  $n$  ( $1 \leq n \leq 8$ ) — количество задач, которые необходимо выполнить.

Следующая строка содержит  $n$  натуральных чисел  $a_i$  ( $1 \leq a_i \leq n$ ) — для каждой работы номер дня, ранее которого не может быть начато выполнение этой задачи. Числа отделены друг от друга одним пробелом.

### Формат выходного файла

В выходной файл `output.txt` выведите одно число — количество различных допустимых расписаний выполнения задач.

### Пример входных и выходных данных

| input.txt      | output.txt |
|----------------|------------|
| 5<br>2 4 4 2 1 | 4          |