

Решение типового варианта для 11 класса

Распределение баллов по заданиям:

Номер задачи	1	2	3	4	5	6
Баллы	5	15	15	18	22	25

Задание 1

Расставьте операции сложения и умножения в строке так, чтобы получилось верное равенство

На вход подаётся строка, содержащая числа, записанные через произвольное число пробелов. Первым числом указывается результат искомого выражения. Результатом ожидается строка, содержащая знаки операций

Входные данные: 24 1 2 3 4

Результат: = * * *

Комментарий: $24 = 1 * 2 * 3 * 4$

Входные данные: 5 3 2

Результат: = +

Комментарий: $5 = 3 * 2$

```
def operations(num, length):
```

```
    res = []
```

```
    for i in range(length):
```

```
        if num % 2 == 0:
```

```
            res.append(0)
```

```
        else:
```

```
            res.append(1)
```

```
            num = num // 2
```

```
    return res[::-1]
```

```
# найдем все числа, которые должны получиться в процессе умножения
```

```
# и перепишем те числа, которые просто складываются
```

```
def count_multiplication(nums_array, ops_array):
```

```
    curr_nums = [nums_array[0]]
```

```
    curr_num_count = 0
```

```
    prev_sum = False
```

```
    for i in range(len(ops_array)):
```

```
        # если умножение
```

```
        if (ops_array[i] == 1):
```

```
            # начало нового этапа умножения, число нужно добавить
```

```
            if (prev_sum):
```

```
                curr_nums.append(nums_array[i] * nums_array[i+1])
```

```

    prev_sum = False
    curr_num_count += 1
# последовательное умножение
else:
    curr_nums[curr_num_count] *= nums_array[i+1]

# если сложение
else:
    prev_sum = True
# последнее число складывается
if i == len(ops_array) - 1:
    curr_nums.append(nums_array[-1])
# будет сумма произведений, чисел добавлять не надо, но надо их разделить
elif (ops_array[i+1] == 1):
    pass
# число "не используется" в умножении, добавим его
else:
    curr_nums.append(nums_array[i])
    curr_num_count += 1

return curr_nums

def test_solution(nums_array, ops_array, awaiting_res):
    res = 0
    mult_result = count_multiplication(nums_array, ops_array)

# оставшиеся числа просто сложим
for elem in mult_result:
    res += elem
    if res > awaiting_res:
        return False

if res == awaiting_res:
    return True
return False

if __name__ == '__main__':
    numbers_arr = input().split()
    for i in range(0, len(numbers_arr)):
        numbers_arr[i] = int(numbers_arr[i])

    resulting_num = numbers_arr[0]
    checked_nums = numbers_arr[1:]

# всего возможных вариантов расстановки знаков
possible_variants = 2 ** (len(checked_nums) - 1)

# каждый из этих вариантов представим как последовательность 0 и 1,
# где 0 - сложение, 1 - умножение
# это можно легко сделать, используя перевод чисел в двоичную систему
variants = []

```

```

for i in range(possible_variants):
    variants.append(operations(i, len(checked_nums) - 1))

solution = []
for variant in variants:
    if (test_solution(checked_nums, variant, resulting_num)):
        solution = variant
        break

result_string = "="
for num in solution:
    if num == 1:
        result_string += " *"
    else:
        result_string += " +"

print(result_string)

```

Задание 2

Из входной строки, содержащей произвольное количество слов (последовательности символов, записанных через пробел), необходимо удалить, сохраняя пробелы, все слова, чья длина равна значению факториала некоего целого числа.

Входные данные: «123 йцукенг й йцу йцукенгшщз зщшг ойойой»

Результат: «123 йцукенг йцу йцукенгшщз зщшг »

Комментарий: длина «й» = 1!, длина «ойойой» = 3!

```

def test_if_factorial(num):
    fact = 1
    curr_step = 2

    while fact < num:
        fact *= curr_step
        curr_step += 1

    if fact == num:
        return True
    return False

if __name__ == '__main__':
    splitted_words = input().split()

    result_string = ""

    for i in range(len(splitted_words)):
        word = splitted_words[i]
        if not(test_if_factorial(len(word))):
            result_string += word

```

```

if i != len(splitted_words) - 1:
    result_string += " "

print(result_string)
#print("--", result_string, "--", sep="")

```

Задание 3

Из входной строки, содержащей произвольное количество слов (последовательности символов, записанных через пробел), необходимо создать строку, удвоив количество пробелов между соседними словами, для которых расстояние Левенштейна меньше 3.

Примечание:

Левенштейна – это минимальное количество операций вставки одного символа, удаления одного символа и замены одного символа на другой, необходимых для превращения одной строки в другую. Например, расстояние Левенштейна между словом «мама» и «папа» будет 2 (2 замены), а между «ель» и «гнёт» – 4 (3 замены и 1 вставка).

```

def lev_distance(str1, str2):
    len1 = len(str1)
    len2 = len(str2)

    if len1 == 1 and len2 == 1:
        if str1 == str2:
            return 0
        else:
            return 1
    elif (len1 == 0) and (len2 > 0):
        return len2
    elif (len2 == 0) and (len1 > 0):
        return len1

    last_sym = 0
    if str1[-1] != str2[-1]:
        last_sym = 1

    return min(lev_distance(str1, str2[:len2-1]) + 1,
               lev_distance(str1[:len1-1], str2) + 1,
               lev_distance(str1[:len1-1], str2[:len2-1]) + last_sym)

if __name__ == '__main__':
    splitted_words = input().split()

    result_string = ""

    for i in range(len(splitted_words) - 1):
        dist = lev_distance(splitted_words[i], splitted_words[i+1])

```

```

result_string += splitted_words[i]
if dist < 3:
    result_string += " "
else:
    result_string += " "

result_string += splitted_words[-1]

print(result_string)

```

Задание 4

На вход подаётся строка, содержащая целые числа. Необходимо найти матрицу, чей определитель будет наибольшим. Результатом программы должна быть строка, содержащая искомую матрицу, записанную построчно.

Входная строка: 1 2 5 6 7

Результат: 7 1 2 6

```

from math import sqrt

def get_all_permutations(nums_array):
    if len(nums_array) == 1:
        return[nums_array]

    res = []
    for i in range(len(nums_array)):
        curr_num = nums_array[i]
        remaining = nums_array[:i] + nums_array[i+1:]

        for elems in get_all_permutations(remaining):
            res.append((curr_num + elems))

    return res

def determinant_recursive(matrix, curr_res):
    size = len(matrix)

    if size == 1:
        return matrix[0][0]

    if size == 2:
        return curr_res * (matrix[0][0] * matrix[1][1] -
                           matrix[1][0] * matrix[0][1])

    # будем брать верхнюю строку матрицы и умножать на детерминант матриц,
    # полученных из остальных строк
    # смотри алгоритм нахождения определителя матрицы N x N на Википедии
    else:
        sign = -1

```

```

sum = 0

for i in range(size):
    add_minor = []
    sign *= -1
    curr_mult = sign * matrix[0][i]

    for j in range(1, size):
        buff = []
        for k in range(size):
            # столбец, в котором выбранный элемент, должен быть убран
            if i != k:
                buff.append(matrix[j][k])
        add_minor.append(buff)

    sum += curr_res * \
        determinant_recursive(add_minor, curr_mult)

return sum

def to_square_matrix(arr, size):
    res = [0] * size
    for i in range(size):
        res[i] = [0] * size

    for i in range(size):
        for j in range(size):
            res[i][j] = arr[i * size + j]

    return res

if __name__ == '__main__':
    numbers_arr = input().split()
    for i in range(0, len(numbers_arr)):
        numbers_arr[i] = int(numbers_arr[i])

    nums_am = len(numbers_arr)

    max_matr_size = int(sqrt(nums_am))

    # найдем все возможные перестановки полученных чисел
    all_permutation = get_all_permutatons(numbers_arr)

    best_matrix = numbers_arr[0]
    best_determinant = numbers_arr[0]

    # от всех возможных перестановок будем "отрезать" нужное нам для создания квадратной
    матрицы
    # число цифр, и будем находить детерминант матрицы, созданной из этих цифр
    for curr_size in range(max_matr_size + 1):
        for elem in all_permutation:
            matr = to_square_matrix(elem, curr_size)

```

```

determ = determinant_recursive(matr, 1)

if determ >= best_determinant:
    best_determinant = determ
    best_matrix = elem[:curr_size*curr_size]

# print(best_determinant)
for elem in best_matrix:
    print(elem, end=" ")

```

Задание 5

На вход подаётся строка, содержащая записанные через пробел целые числа. Необходимо найти наибольшую сумму чисел, расположенных между числами, имеющими индексы, равные соседним числам Фибоначчи (граничные числа в сумме учитываются).

Примечание: числа Фибоначчи – последовательность чисел, где следующее число равно сумме двух предыдущих: 0,1,1,2,3,5,8 ...

Входная строка: 8 2 4 3 9 1 2 3 4 5

Результат: 13

Комментарий к ответу: [3;5]3е число это 3, 5е число это 1, 3+9+1 = 13; [5;8] = 1+2+3+4=10

```

# создадим массив из чисел Фибоначчи, которые не превышают размер массива
# чтобы использовать для индексации
def create_Fibonacci_arr(nums_amount):
    arr = [0, 1]
    counter = 1
    i = 1
    while i <= nums_amount - 1:
        arr.append(i)
        counter += 1
        i = arr[counter] + arr[counter - 1]

    return arr

if __name__ == '__main__':

    numbers_arr = input().split()
    for i in range(0, len(numbers_arr)):
        numbers_arr[i] = int(numbers_arr[i])

    nums_amount = len(numbers_arr)

    fibonacci_arr = create_Fibonacci_arr(nums_amount)
    # print(fibonacci_arr)

    # первые числа Фибоначчи - 0 и 1, поэтому начальное значение
    # максимальной суммы элементов - сумма 0-го и 1-го элемента введенного массива
    max_sum = numbers_arr[fibonacci_arr[0]] + numbers_arr[fibonacci_arr[1]]

```

```

# последовательно берем пары элементов из массива чисел Фибоначчи
# и суммируем элементы введенного массива, используя эти пары как индексы
# полученную сумму сравниваем с текущей максимальной
for i in range(1, len(fibonacci_arr) - 1):
    curr_sum = 0
    for j in range(fibonacci_arr[i], fibonacci_arr[i+1] + 1):
        curr_sum += numbers_arr[j]

    if curr_sum > max_sum:
        max_sum = curr_sum

print(max_sum)

```

Задание 6

На вход подаётся строка, содержащая записанные через пробел целые числа. Необходимо найти наибольшее количество чисел, сумма цифр которых является одинаковым простым числом.

Входная строка: 19 23 113 302 104 3 12

Результат: 4

```

from math import sqrt

# найдем сумму цифр числа
# будем добавлять цифры "с конца", используя остаток от деления на 10
# после чего "отрежем" эту цифру с помощью целочисленного деления
# (другой вариант решения подзадачи - представить число как строку и
# переводить каждый char в цифру отдельно)
def sum_of_digits(num):
    sum = 0

    while num > 0:
        sum += num % 10
        num = num // 10
    return sum

# используем "топорный" алгоритм, так как максимальная оптимизация не требуется
# будем проверять делимость числа на все целые числа, меньше корня этого числа,
# хотя достаточно проверять делимость только на меньшие простые
def test_if_prime(num):
    for i in range(2, round(sqrt(num)) + 1):
        a = num / i
        b = num // i
        if a == b:
            return False
    return True

if __name__ == '__main__':
    numbers_arr = input().split()
    for i in range(0, len(numbers_arr)):

```

```

numbers_arr[i] = int(numbers_arr[i])

# создадим массив, где будем записывать встреченные нами простые суммы цифр
prime_sums = []
for curr_number in numbers_arr:
    curr_digit_sum = sum_of_digits(curr_number)
    if (test_if_prime(curr_digit_sum)):
        prime_sums.append(curr_digit_sum)
# print(prime_sums)

max_prime_count = 0
# посчитаем кол-во вхождений каждого из простых чисел, чтобы найти наибольшее
# для простоты даже не будем проверять, считали ли мы уже сумму вхождений
# для этого числа
for curr_number in prime_sums:
    count = 0
    for checked_number in prime_sums:
        if curr_number == checked_number:
            count += 1

    if count > max_prime_count:
        max_prime_count = count

print(max_prime_count)

```

**Заключительный (очный) этап академического соревнования
Олимпиады школьников «Шаг в будущее» по программированию, весна 2019 г.
11 класс**

Вариант 1

Задача 1

Эльф, гном и человек шли по подземелью в поисках сокровищ. Упервшись в стену, которая заграждала проход:

- гном пришёл к выводу, что это не стена, а дверь, которую заложили мастера древности, а за дверью их ждёт сокровище;
- эльф припомнил, что мастера древности любили загадки и математические равенства;
- человек узнал алфавит используемой древними системы счисления.

После вынужденного обсуждения также были выяснены следующие факты:

- на стену нанесены числа в некой позиционной системе, причём операция находится между аргументами;
- основание системы счисления целое и положительное;
- для ответа нужно указать ответ в используемой системе счисления.

На вход подаётся строка с основанием используемой системы счисления и перечислением элементов алфавита, а также выражение, к которому нужно указать ответ. Ответ дать в указанной системе счисления в формате строки.

Входная строка представляет собой число-основание системы счисления и последовательность разрядов, записанных через произвольное число пробелов по возрастанию значения, после чего – строковая запись выражения.

Выходной строкой является строка, содержащая ответ, который необходимо найти по заданию.

Изменять формат входных и выходных данных запрещено.

Оставлять в конце программы ожидание ввода запрещено.

Помните, что автоматическая проверка населена роботами.

Входная строка: 4 0 1 2 3 11+10

Результат: 21

Критерии: Простая задача – 5 баллов

N	Оценка	Входные данные	Выходные данные
1	1	4 0 1 2 3 11+10	21
2	1	5 0 1 2 3 4 31+20	101

3	1	3 0 1 2 21+20	111
4	1	7 0 1 2 3 4 5 6 51+50	131
5	1	5 a b c d e db+ca	bab

Решение:

```
#include "stdafx.h"
```

```
#include <iostream>
```

```
#include <string>
```

```
#include <cmath>
```

```
std::string convertSingleTo(int num, char *symbols)
{
    return std::string() + symbols[num];
}
```

```
std::string convertTo(int num, int base, char *symbols)
{
    if(abs(num) < base) return convertSingleTo(abs(num), symbols);
    if(num < 0)
        return "-" + (convertTo(abs(num)/base, base, symbols) + convertSingleTo(abs(num)%base,
symbols));
    return convertTo(abs(num)/base, base, symbols) + convertSingleTo(abs(num)%base, symbols);
}
```

```
int convertSingleFrom(char num, char *symbols)
{
    int i=0;
    while(symbols[i] != num) i++;
    return i;
}
```

```
int convertFrom(std::string num, int base, char *symbols)
{
    int k=1;
    if(num[0] == '-')
    {
        k = -1;
        num.erase(num.begin());
    }
    int result = 0;
    for(int i=num.length()-1; i>=0; i--)
        result += convertSingleFrom(num[i], symbols)*pow(base, num.length()-1-i);
    return k*result;
}
```

```
int findOperandPos(std::string str)
{
    char *operands = new char[5];
    operands[0] = '+';
```

```

operands[1] = '-';
operands[2] = '*';
operands[3] = '/';
operands[4] = '%';
for(int i=0; i<str.length(); i++)
    for(int k=0; k<5; k++)
        if(str[i] == operands[k]) return i;
return -1;
}

std::string doTask(std::string task, int base, char *symbols)
{
    for(int i=task.length(); i>=0; i--)
        if((task[i] == ')') || (task[i] == '('))
            task.erase(task.begin() + i);

    int first = 1;
    if(task[0] == '-')
    {
        task.erase(task.begin());
        first = -1;
    }

    int operand = findOperandPos(task);
    first *= convertFrom(task.substr(0, operand), base, symbols);

    task.erase(task.begin(), task.begin() + operand);

    int second = 1;
    if(task[1] == '-')
    {
        task.erase(task.begin() + 1);
        second = -1;
    }
    second *= convertFrom(task.substr(1, task.length()-1), base, symbols);

    int result;
    switch(task[0])
    {
        case '+': result = first+second; break;
        case '-': result = first-second; break;
        case '*': result = first*second; break;
        case '/': result = first/second; break;
        case '%': result = first%second; break;
        default: result = -1; break;
    }

    return convertTo(result, base, symbols);
}

int main()
{

```

```

int base;
std::cin>>base;
char *symbols = new char[base];
for(int i=0; i<base; i++)
    std::cin>>symbols[i];
std::string task;
std::cin>>task;

std::string result = doTask(task, base, symbols);
std::cout<<result<<"\n";
return 0;
}

```

Задача 2

Эльф, гном и человек шли по подземелью в поисках сокровищ. Уперевшись в стену, которая заграждала проход:

- гном пришёл к выводу, что это не стена, а дверь, которую заложили мастера древности, а за дверью их ждёт сокровище;
- человек нашёл на полу невообразимое количество фиолетовых, синих, жёлтых и красных камешков, но лишь один зелёный;
- эльф припомнил, что мастера древности любили загадки и математические равенства;

После вынужденного обсуждения также были выяснены следующие факты:

- на стену нанесены барельефы чисел;
- между числами находятся пазы (куда успешно помещаются найденные разноцветные камешки – по одному в паз);
- мастера древности рассматривали цвета как знаки арифметических операций:
 - сложение – фиолетовым;
 - умножение – красным;
 - вычитание – синим;
 - равенство – зелёным;
 - целочисленное деление – жёлтым;
 - остаток от деления – оранжевым;
- древняя культура использовала только целые числа.

На вход программе подаётся строка с целыми числами, нанесённых на стену, расставьте камни так, чтобы получилось математическое равенство. Ответ запишите строкой, указывая через пробел символы (+, -, *, / – целочисленное деление, % – остаток от деления; операции указаны в порядке убывания частоты использования) арифметических операций, которые символизируются нужными

цветами. В случае, если таких записей несколько укажите вариант с наиболее часто употребляемыми операциями.

Входная строка представляет собой последовательность чисел, записанные через произвольное число пробелов.

Выходной строкой является строка, которую необходимо найти по заданию.

Изменять формат входных и выходных данных запрещено.

Оставлять в конце программы ожидание ввода запрещено.

Помните, что автоматическая проверка населена роботами.

Входные данные: 5 2 14 7

Результат: + = -

Критерии: простая 15 баллов

N	Оценка	Входные данные	Выходные данные
1	1	5 2 14 7	+ = -
2	1	9 2 7 3 3	- - + =
3	1	2 9 9 9	* = +
4	1	3 6 10 8	* = +
5	5	1 2 3 2 17 8	+ + + % =
6	1	6 9 3	= -
7	5	81 27 9 0 6	/ - = -

Решение:

```
s=list(map(int, input().split()))
n=len(s)
slf={0:"+", 1:"- ", 2:"*", 3:"/", 4:"%", 5:"==", 6:"="}
a=[0]*(n-1)
```

```
f=0
```

```
while ( sum(a)<4*(n-1) and f==0):
```

```
    for j in range(n-1):
```

```
        q=a[j]
```

```
        a[j]=5
```

```
        ans=""
```

```
        for i in range(n-1):
```

```
            ans+=str(s[i])+slf[a[i]]
```

```
        ans+=str(s[-1])
```

```
    if eval(ans) and f==0:
```

```
        f=1
```

```
        for k in range(n-1):
```

```
            w=a[k]
```

```
            if w==5:
```

```

    w+=1
    print(slf[w], end=" ")
print()

```

a[j] = q

```

o=1
for i in range(n-1):
    a[i]+=o
    o=a[i]//5
    a[i]=a[i]%5

```

Задача 3

При анализе генома человека иногда есть необходимость находить повторяющиеся последовательности в строке, состоящей из алфавита А,С,Г,Т. Необходимо составить алгоритм, работающий для произвольного генома, определяющий в заданной строке наиболее длинную подстроку, которая повторяется заданное количество раз (или вернуть пустую строку).

Входная строка представляет собой число повторений и заданная строка, записанные через произвольное число пробелов.

Выходной строкой является строка, которую необходимо найти по заданию.

Изменять формат входных и выходных данных запрещено.

Оставлять в конце программы ожидание ввода запрещено.

Помните, что автоматическая проверка населена роботами.

Входная строка: 3 ACGCCGCTTTTCCGGTTCGC.

Результат: CGC

Пояснение: последовательность ТТ тоже подойдёт, но она более короткая, чем CGC.

Критерии: средняя 15 баллов

N	Оценка	Входные данные	Выходные данные
1	1	3 ACGCCGCTTTTCCGGTTCGC	CGC
2	1	3 ухогорлоносоговорилородогурцов	ого
3	1	3 inhfa pf ghtdsitybt crjhjcnb cnj, inhfa pf gmzycndj ldf, inhfa pf ghjcnj nfr vyjuj	inhfa pf
4	1	2 перпендикуляр, опущенный из заданной точки К на плоскость, будет также параллелен прямой АС, которая вместе с заданной точной К образует плоскость.	плоскость
5	5	1000 На краю дороги стоял дуб. Он был, вероятно, в десять раз старше берез, составлявших лес, в десять раз толще и в два раза выше каждой березы. Это был огромный, в два обхвата дуб, с обломанными суками и корой, заросшей старыми болячками. С огромными, неуклюже, несимметрично растопыренными корявыми руками и пальцами, он	

		старым, сердитым и презрительным уродом стоял между улыбающимися березами.	
6	5	0	
7	1	4 ехать на конференцию на верхней полке на две недели на поезде	на

Решение:

```

var
  n,i,i2,i3,c:longint;
  s,current: string;
  checked: array[1..100000] of string;
  function rec(st: string;depth: longint; j: longint):boolean;
  begin
    if not (st[depth]=s[j]) then begin rec:= false; exit; end;
    if depth = length(st) then begin rec:= true; exit; end else rec:=rec(st,depth+1,j+1);

  end;

  function count(str: string):longint;
  var
    j,cou,ans: longint;
  begin
    ans:=0;
    j:=1;
    for j:=1 to length(s) do
      begin
        if str[1]=s[j] then if rec(str,1,j) then ans:=ans+1;
        end;
        count:=ans;
      end;
    end;

  procedure check(str: string);
  var
    j: longint;
  begin
    for j:=1 to c do
      if checked[j]=str then exit;
      c:=c+1;
      checked[c]:=str;
      if count(str)=n then begin writeln(str); halt(); end;
    end;

  begin
  read(n);
  readln(s);
  c:=0;
  //      n:=2;
  //  s:='BCCBBCCBCBCCBC';
  current:="";
  for i:=length(s) downto 1 do
  for i2:=1 to length(s)-i do
  begin

```

```

for i3:=i2 to i2+i-1 do
begin
  current:=current+s[i3];
end;
check(current);
current:="";
end;
end.

```

Задача 4

Вася и Коля поспорили, что если особым образом связать шнурки от ботинок у всего класса, то можно получить две верёвки одинаковой длины. Однако Вася не учёл, что не у всех присутствующих шнурки одинаковой длины. Укажите длины верёвок, которые можно получить при связывании, так, чтобы их длины были примерно одинаковы (размером узлов пренебречь).

На вход подаётся строка с длинами шнурков (только целые числа), записанных через пробел.

На выходе ожидается строка с двумя длинами, записанными через пробел в порядке убывания длины.

Изменять формат входных и выходных данных запрещено.

Оставлять в конце программы ожидание ввода запрещено.

Помните, что автоматическая проверка населена роботами.

Входные параметры	Выходные параметры
5 1 100 20 40 30 50	125 121
1 10	10 1

Критерии: средняя 18 баллов

N	Оценка	Входные данные	Выходные данные
1	5	5 1 100 20 40 30 50	125 121
2	5	1 10	10 1
3	8	5 10 95 75 100 115 20	210 210

Решение:

```

program Project4;
  type mas=array[1..100] of integer;
  var s2,s1:string; m:mas; i,x,y, n,c:integer; f:boolean;
  { procedure rec(s1,s2:string;x,y:integer) ;
  var a,c:integer; s3:string;
  begin
    if length(s1)=0 then
    if x=y then readln(s2);
    if length(s1)>0 then
    begin
      s3:=copy(s1,pos(' ',s1)+1,length(s1)-pos(' ',s1));

```

```

val(copy(s1,1,pos(' ',s1)-1),a,c);
if(length(s2)=0) then rec(s3,s2,a,y)
else
  begin
    rec(s3,s2+"+",x+a,y);
    rec(s3,s2+"-",x-a,y);
    rec(s3,s2+"*",x*a,y);
    rec(s3,s2+"/",x/a,y);
  end;

end;
end;
}
function sum(x,y:integer;m:mas):integer;
var i,s:integer;
begin
  s:=0;
  for i:= x to y do
    s:= s+m[i];
  sum:=s;
end;
begin
  readln(s1);
  n:=1;
  f:=true;
  while(pos(' ',s1)>0) do
    begin
      val(copy(s1,1,pos(' ',s1)-1),m[n],c);
      delete(s1,1,pos(' ',s1));
      n:=n+1;
    end;
    val(s1,m[n],c);

  while(f) do
    begin
      f:=false;
      for i:= 1 to n-1 do
        if m[i]<m[i+1] then
          begin f:= true;c:= m[i];m[i]:=m[i+1];m[i+1]:=c;end;
        end;
      for i:= 1 to n do
        begin
          if x>y then y:=y+m[i]
          else x:=x+ m[i];
        end;
      if x>y then writeln(x,' ',y)
      else writeln(y,' ',x) ;
    end.

```

Задача 5

В классе размещаются N рядов по M парт в каждом. За каждой партой сидит 1 школьник. Необходимо рассадить детей так, чтобы выполнялись условия:

- отдельно школьников можно перемещать только между рядами;
- самые высокие школьники должны сидеть на левом (нулевом) ряду, а самые низкие – на правом;
- парты, равноудалённые от доски, должны быть заняты с повышением среднего роста сидящих там школьников;

На вход подаётся матрица, где каждая ячейка – рост школьника. На выходе – изменённая матрица. Входная строка представляет собой последовательность чисел, записанные через произвольное число пробелов. Первые два числа – количество столбцов и строк.

Выходной строкой является строка с числами, записанными через пробел, которую необходимо найти по заданию.

Изменять формат входных и выходных данных запрещено.

Оставлять в конце программы ожидание ввода запрещено.

Помните, что автоматическая проверка населена роботами.

Входная строка: 6 3 9 5 8 3 7 4 6 1 2 1 9 7 4 3 6 2 5 5

Выходная строка 6 5 5 4 3 2 9 7 6 2 1 1 9 8 7 5 4 3

Критерии: сложная 22 балла

N	Оценка	Входные данные	Выходные данные
1	7	6 3 9 5 8 3 7 4 6 1 2 1 9 7 4 3 6 2 5 5	6 5 5 4 3 2 9 7 6 2 1 1 9 8 7 5 4 3
2	7	3 3 9 3 5 1 2 6 2 3 4	6 2 1 4 3 2 9 5 3
3	8	0 0	

Решение:

```
l = list(map(int,input().split()))

n,m = l[0],l[1]
l = l[2:]
#n,m = map(int, input().split())
a = []
for i in range(m):
    a.append(l[:n])
    l = l[n:]
#a.append(list(map(int,input().split())))
a[i].sort()
a[i].reverse()
for i in range(len(a)):
    for j in range(i,len(a)):
        if j>i and sum(a[i])>sum(a[j]):
            arr = a[i]
            a[i] = a[j]
```

```

a[j] = arr
ans = []
for i in range(m):
    ans = ans+a[i]
print(*ans)

```

Задача 6

Леший борется с незаконной вырубкой леса: он спутывает ветви деревьев так, что дерево, которое срубают, держится кроной за соседние деревья и не падает. Таким образом, чтобы свалить одно дерево нужно срубить все деревья, за которые держится первое дерево. Также, пока никто не видит, леший может переставлять деревья поближе друг к другу.

На вход подаётся строка с количеством ветвей, способных к соединению, укажите какое наибольшее число деревьев может быть соединено в единую структуру?

Входная строка представляет собой последовательность целых чисел, записанных через произвольное число пробелов.

Выходной строкой является строка, содержащая число, которое необходимо найти по заданию.

Изменять формат входных и выходных данных запрещено.

Оставлять в конце программы ожидание ввода запрещено.

Помните, что автоматическая проверка населена роботами.

Вход: 3 1 1 1 1 1 1 1 1 2 1 1 1 1

Выход: 5

Критерии: сложная 25 баллов

N	Оценка	Входные данные	Выходные данные
1	1	3 1 1 1 1 1 1 1 1 2 1 1 1 1	5
2	1	3 1 1 1 1 1 1 5 1 1 1 1 1 1 1 1 1 1	8
3	1	1 2 2 2 1 1 1 1 1 1 1 1 1 1 1 4	8
4	1	3 2 1 1 1 1 1 2 1	6
5	8	9 5	2
6	7	2 2 2 2 2 2 2 8	8
7	6		0

Решение:

```

a = input().split()
rez = 0
lenth = 0
notfind = 1
for i in a:
    i = int(i)
    if notfind:
        rez += i+1
        notfind = 0

```

```
else:  
    rez += i-1  
    lenth += 1  
print(min(rez,lenth))
```

**Заключительный (очный) этап академического соревнования
Олимпиады школьников «Шаг в будущее» по программированию, весна 2019 г.
11 класс**

Вариант 2

Задача 1

При контакте с инопланетянами потребовалось договориться о выборе способа записи математических выражений (из-за большого количества различных народностей, каждая из которых использует свои обозначения):

- использование позиционной системы счисления;
- использование инфиксной записи и приоритетами операций, принятых у людей (умножение и деление имеют более высокий приоритет, чем сложение и вычитание);
- каждый раз необходимо явно указывать, какое основание (целое, положительное) системы счисления используется, и какой алфавит эта система счисления имеет;

На вход подаётся строка с основанием используемой системы счисления и перечислением элементов алфавита, а также выражение, к которому нужно указать ответ. Ответ дать в указанной системе счисления в формате строки.

Входная строка представляет собой число-основание системы счисления и последовательность разрядов, записанных через произвольное число пробелов по возрастанию значения, после чего – строковая запись выражения.

Выходной строкой является строка, содержащая ответ, который необходимо найти по заданию.

Изменять формат входных и выходных данных запрещено.

Оставлять в конце программы ожидание ввода запрещено.

Помните, что автоматическая проверка населена роботами.

Входная строка: 5 0 1 2 3 4 31+20

Результат: 101

Критерии: Простая задача – 5 баллов

N	Оценка	Входные данные	Выходные данные
1	1	4 0 1 2 3 11+10	21
2	1	5 0 1 2 3 4 31+20	101
3	1	3 0 1 2 21+20	111
4	1	7 0 1 2 3 4 5 6 51+50	131
5	1	5 a b c d e db+ca	bab

Решение:

```

#include "stdafx.h"

#include <iostream>
#include <string>
#include <cmath>

std::string convertSingleTo(int num, char *symbols)
{
    return std::string() + symbols[num];
}

std::string convertTo(int num, int base, char *symbols)
{
    if(abs(num) < base) return convertSingleTo(abs(num), symbols);
    if(num < 0)
        return "-" + (convertTo(abs(num)/base, base, symbols) + convertSingleTo(abs(num)%base,
symbols));
    return convertTo(abs(num)/base, base, symbols) + convertSingleTo(abs(num)%base, symbols);
}

int convertSingleFrom(char num, char *symbols)
{
    int i=0;
    while(symbols[i] != num) i++;
    return i;
}

int convertFrom(std::string num, int base, char *symbols)
{
    int k=1;
    if(num[0] == '-')
    {
        k = -1;
        num.erase(num.begin());
    }
    int result = 0;
    for(int i=num.length()-1; i>=0; i--)
        result += convertSingleFrom(num[i], symbols)*pow(base, num.length()-1-i);
    return k*result;
}

int findOperandPos(std::string str)
{
    char *operands = new char[5];
    operands[0] = '+';
    operands[1] = '-';
    operands[2] = '*';
    operands[3] = '/';
    operands[4] = '%';
    for(int i=0; i<str.length(); i++)
        for(int k=0; k<5; k++)
            if(str[i] == operands[k]) return i;
}

```

```

        return -1;
    }

std::string doTask(std::string task, int base, char *symbols)
{
    for(int i=task.length(); i>=0; i--)
        if((task[i] == ')') || (task[i] == '('))
            task.erase(task.begin() + i);

    int first = 1;
    if(task[0] == '-')
    {
        task.erase(task.begin());
        first = -1;
    }

    int operand = findOperandPos(task);
    first *= convertFrom(task.substr(0, operand), base, symbols);

    task.erase(task.begin(), task.begin() + operand);

    int second = 1;
    if(task[1] == '-')
    {
        task.erase(task.begin() + 1);
        second = -1;
    }
    second *= convertFrom(task.substr(1, task.length()-1), base, symbols);

    int result;
    switch(task[0])
    {
        case '+': result = first+second; break;
        case '-': result = first-second; break;
        case '*': result = first*second; break;
        case '/': result = first/second; break;
        case '%': result = first%second; break;
        default: result = -1; break;
    }

    return convertTo(result, base, symbols);
}

int main()
{
    int base;
    std::cin>>base;
    char *symbols = new char[base];
    for(int i=0; i<base; i++)
        std::cin>>symbols[i];
    std::string task;
    std::cin>>task;
}

```

```
std::string result = doTask(task, base, symbols);
std::cout<<result<<"\n";
return 0;
}
```

Задача 2

Эльф, гном и человек шли по подземелью в поисках сокровищ. Уперевшись в стену, которая заграждала проход:

- гном пришёл к выводу, что это не стена, а дверь, которую заложили мастера древности, а за дверью их ждёт сокровище;
- человек нашёл на полу невообразимое количество фиолетовых, синих, жёлтых и красных камешков, но лишь один зелёный;
- эльф припомнил, что мастера древности любили загадки и математические равенства;

После вынужденного обсуждения также были выяснены следующие факты:

- на стену нанесены барельефы чисел;
- между числами находятся пазы (куда успешно помещаются найденные разноцветные камешки – по одному в паз);
- мастера древности рассматривали цвета как знаки арифметических операций:
 - сложение – фиолетовым;
 - умножение – красным;
 - вычитание – синим;
 - равенство – зелёным;
 - целочисленное деление – жёлтым;
 - остаток от деления – оранжевым;
- древняя культура использовала только целые числа.

На вход программе подаётся строка с целыми числами, нанесённых на стену, расставьте камни так, чтобы получилось математическое равенство. Ответ запишите строкой, указывая через пробел символы (+, -, *, / – целочисленное деление, % – остаток от деления; операции указаны в порядке убывания частоты использования) арифметических операций, которые символизируются нужными цветами. В случае, если таких записей несколько укажите вариант с наиболее часто употребляемыми операциями.

Входная строка представляет собой последовательность чисел, записанные через произвольное число пробелов.

Выходной строкой является строка, которую необходимо найти по заданию.

Изменять формат входных и выходных данных запрещено.

Оставлять в конце программы ожидание ввода запрещено.

Помните, что автоматическая проверка населена роботами.

Входные данные: 5 2 14 7

Результат: + = -

Критерии: простая 15 баллов

N	Оценка	Входные данные	Выходные данные
1	1	5 2 14 7	+ = -
2	1	9 2 7 3 3	- - + =
3	1	2 9 9 9	* = +
4	1	3 6 10 8	* = +
5	5	1 2 3 2 17 8	+ + + % =
6	1	6 9 3	= -
7	5	81 27 9 0 6	/ - = -

Решение:

```
s=list(map(int, input().split()))
n=len(s)
slf={0:"+", 1:"- ", 2:"*", 3:"/ ", 4:"% ", 5:"==", 6:"="}
a=[0]*(n-1)
```

```
f=0
```

```
while ( sum(a)<4*(n-1) and f==0):
```

```
    for j in range(n-1):
```

```
        q=a[j]
```

```
        a[j]=5
```

```
        ans=""
```

```
        for i in range(n-1):
```

```
            ans+=str(s[i])+slf[a[i]]
```

```
        ans+=str(s[-1])
```

```
    if eval(ans) and f==0:
```

```
        f=1
```

```
        for k in range(n-1):
```

```
            w=a[k]
```

```
            if w==5:
```

```
                w+=1
```

```
            print(slf[w], end=" ")
```

```
        print()
```

```
    a[j] = q
```

```
o=1
```

```

for i in range(n-1):
    a[i]+=o
    o=a[i]/5
    a[i]=a[i]%5

```

Задача 3

В центре отслеживания внеземных цивилизаций заметили повторяющийся сигнал. После перевода этого сигнала в последовательность символов, «исследователи» предположили, что для расшифровки сигнала им необходимо найти повторяющиеся подпоследовательности. Ориентируясь на закон Ципфа, пришли к выводу, что нужно выбирать не самую короткую, и не самую длинную подпоследовательность. Необходимо найти наиболее длинную последовательность слов, которая повторяется в этой строке выбранное число раз.

Входная строка представляет собой число повторений и заданная строка, записанные через произвольное число пробелов.

Выходной строкой является строка, которую необходимо найти по заданию.

Изменять формат входных и выходных данных запрещено.

Оставлять в конце программы ожидание ввода запрещено.

Помните, что автоматическая проверка населена роботами.

Входная строка: 3 inhfa pf ghtdsitybt crjhjcnb cnj, inhfa pf gmzycndj ldf, inhfa pf ghjcnj nfr vyuj.

Результат: «inhfa pf »

Пояснение: 3 – заданное количество повторений, после него идёт строка.

Критерии: средняя 15 баллов

N	Оценка	Входные данные	Выходные данные
1	1	3 ACGCCGCTTTCCGGTTCGC	CGC
2	1	3 ухогорлоносоговорилородогурцов	ого
3	1	3 inhfa pf ghtdsitybt crjhjcnb cnj, inhfa pf gmzycndj ldf, inhfa pf ghjcnj nfr vyuj	inhfa pf
4	1	2 перпендикуляр, опущенный из заданной точки K на плоскость, будет также параллелен прямой AC, которая вместе с заданной точкой K образует плоскость.	плоскость
5	5	1000 На краю дороги стоял дуб. Он был, вероятно, в десять раз старше берез, составлявших лес, в десять раз толще и в два раза выше каждой березы. Это был огромный, в два обхвата дуб, с обломанными суками и корой, заросшей старыми болячками. С огромными, неуклюже, несимметрично растопыренными корявыми руками и пальцами, он старым, сердитым и презрительным уродом стоял между улыбающимися березами.	
6	5	0	
7	1	4 ехать на конференцию на верхней полке на две недели на поезде	на

Решение:

```
var
  n,i,i2,i3,c:longint;
  s,current: string;
  checked: array[1..100000] of string;
function rec(st: string;depth: longint; j: longint):boolean;
begin
  if not (st[depth]=s[j]) then begin rec:= false; exit; end;
  if depth = length(st) then begin rec:= true; exit; end else rec:=rec(st,depth+1,j+1);

end;

function count(str: string):longint;
var
  j,cou,ans: longint;
begin
  ans:=0;
  j:=1;
  for j:=1 to length(s) do
  begin
    if str[1]=s[j] then if rec(str,1,j) then ans:=ans+1;
  end;
  count:=ans;
end;

procedure check(str: string);
var
  j: longint;
begin
  for j:=1 to c do
  if checked[j]=str then exit;
  c:=c+1;
  checked[c]:=str;
  if count(str)=n then begin writeln(str); halt(); end;
end;

begin
read(n);
readln(s);
  c:=0;
//      n:=2;
//  s:='BCCBCCBCBCCBC';
  current:="";
  for i:=length(s) downto 1 do
  for i2:=1 to length(s)-i do
  begin
    for i3:=i2 to i2+i-1 do
    begin
      current:=current+s[i3];
    end;
    check(current);
  end;
end;
```

```

current:="";
end;
end.

```

Задача 4

Можно купить 2 тягача по цене одного для транспортировки грузов. Определите тяговую способность одного тягача, чтобы нагрузка была примерно одинакова для обоих тягачей.

На вход подаётся строка с весами грузов, которые нужно доставить по железной дороге, записанных через пробел.

На выходе ожидается строка с двумя числами, записанными через пробел в порядке убывания, соответствующие тяговой нагрузке на каждый из сформированных составов.

Изменять формат входных и выходных данных запрещено.

Оставлять в конце программы ожидание ввода запрещено.

Помните, что автоматическая проверка населена роботами.

Входные параметры	Выходные параметры
5 1 100 20 40 30 50	125 121
1 10	10 1

Критерии: средняя 18 баллов

N	Оценка	Входные данные	Выходные данные
1	5	5 1 100 20 40 30 50	125 121
2	5	1 10	10 1
3	8	5 10 95 75 100 115 20	210 210

Решение:

```

program Project4;
  type mas=array[1..100] of integer;
  var s2,s1:string; m:mas; i,x,y, n,c:integer; f:boolean;
  { procedure rec(s1,s2:string;x,y:integer) ;
  var a,c:integer; s3:string;
  begin
    if length(s1)=0 then
    if x=y then readln(s2);
    if length(s1)>0 then
    begin
      s3:=copy(s1,pos(' ',s1)+1,length(s1)-pos(' ',s1));
      val(copy(s1,1,pos(' ',s1)-1),a,c);
      if(length(s2)=0) then rec(s3,s2,a,y)
      else

```

```

begin
  rec(s3,s2+"+",x+a,y);
  rec(s3,s2+"-",x-a,y);
  rec(s3,s2+"*",x*a,y);
  rec(s3,s2+"/",x/a,y);
end;

end;
end;
}
function sum(x,y:integer;m:mas):integer;
var i,s:integer;
begin
  s:=0;
  for i:= x to y do
    s:= s+m[i];
  sum:=s;
end;
begin
  readln(s1);
  n:=1;
  f:=true;
  while(pos(' ',s1)>0) do
  begin
    val(copy(s1,1,pos(' ',s1)-1),m[n],c);
    delete(s1,1,pos(' ',s1));
    n:=n+1;
  end;
  val(s1,m[n],c);

  while(f) do
  begin
    f:=false;
    for i:= 1 to n-1 do
    if m[i]<m[i+1] then
    begin f:= true;c:= m[i];m[i]:=m[i+1];m[i+1]:=c;end;
    end;
    for i:= 1 to n do
    begin
    if x>y then y:=y+m[i]
    else x:=x+ m[i];
    end;
    if x>y then writeln(x,' ',y)
    else writeln(y,' ',x) ;
  end.

```

Задача 5

Римские манипулы состояли из людей, построенных в виде прямоугольника со сторонами М и N. Для повышения боеспособности манипулы заполнялись по увеличению опыта ряда (чем меньше опыта у членов ряда, тем ближе этот ряд к передовой), а распределение внутри ряда – по уменьшению опыта слева направо (на 0й позиции стоит человек с наибольшим опытом, далее по не возрастанию до крайней позиции). В легион прибывают отшагавшие новобранцы, построившиеся манипулой. Центурион не хочет понижать боевой дух, разделяя успешных сдружиться членов ряда. Помогите центуриону расположить легионеров, согласно требованиям.

На вход подаётся матрица, где каждая ячейка – имеющийся опыт. На выходе – изменённая матрица в виде строки.

Входная строка представляет собой последовательность чисел, записанные через произвольное число пробелов. Первые два числа – количество столбцов и строк.

Выходной строкой является строка с числами, записанными через пробел, которую необходимо найти по заданию.

Изменять формат входных и выходных данных запрещено.

Оставлять в конце программы ожидание ввода запрещено.

Помните, что автоматическая проверка населена роботами.

Входная строка: 3 3 9 3 5 1 2 6 2 3 4

Результат: 6 2 1 4 3 2 9 5 3

Критерии: сложная 22 баллов

N	Оценка	Входные данные	Выходные данные
1	7	6 3 9 5 8 3 7 4 6 1 2 1 9 7 4 3 6 2 5 5	6 5 5 4 3 2 9 7 6 2 1 1 9 8 7 5 4 3
2	7	3 3 9 3 5 1 2 6 2 3 4	6 2 1 4 3 2 9 5 3
3	8	0 0	

Решение:

```
l = list(map(int,input().split()))

n,m = l[0],l[1]
l = l[2:]
#n,m = map(int, input().split())
a = []
for i in range(m):
    a.append(l[:n])
    l = l[n:]
    #a.append(list(map(int,input().split())))
    a[i].sort()
    a[i].reverse()
for i in range(len(a)):
    for j in range(i,len(a)):
```

```

if j>i and sum(a[i])>sum(a[j]):
    arr = a[i]
    a[i] = a[j]
    a[j] = arr
ans = []
for i in range(m):
    ans = ans+a[i]
print(*ans)

```

Задача 6

Наутро после того, как в четверг прошёл дождь, вызвавший незапланированное посвистывание у всех горных раков, проснулось волшебное существо Еинелсичто, которое решило раз и навсегда решить вопрос должников по учебным дисциплинам. Ночью поместило оно всех студентов, не учивших, но имеющих задолженности по различным предметам, в магическую коробку и наложило страшное заклинание: каждый попавший под заклинание притягивал к себе столько собратьев по несчастью, сколько задолженностей он имел (притяжение взаимное).

На вход подаётся строка с количеством задолженностей для каждого студента. Определите наибольшее число студентов, объединённых притяжением заклинания, в одной получившейся «кучке».

Входная строка представляет собой последовательность целых чисел, записанных через произвольное число пробелов.

Выходной строкой является строка, содержащая число, которое необходимо найти по заданию.

Изменять формат входных и выходных данных запрещено.

Оставлять в конце программы ожидание ввода запрещено.

Помните, что автоматическая проверка населена роботами.

Вход: 1 2 2 2 1 1 1 1 1 1 1 1 1 1 4

Выход: 8

Критерии: сложная 25 баллов

N	Оценка	Входные данные	Выходные данные
1	1	3 1 1 1 1 1 1 1 1 2 1 1 1 1	5
2	1	3 1 1 1 1 1 1 5 1 1 1 1 1 1 1 1 1	8
3	1	1 2 2 2 1 1 1 1 1 1 1 1 1 1 1 4	8
4	1	3 2 1 1 1 1 1 2 1	6
5	8	9 5	2
6	7	2 2 2 2 2 2 2 8	8
7	6		0

Решение:

```

a = input().split()
rez = 0

```

```
lenth = 0
notfind = 1
for i in a:
    i = int(i)
    if notfind:
        rez += i+1
        notfind = 0
    else:
        rez += i-1
    lenth += 1
print(min(rez,lenth))
```

**Заключительный (очный) этап академического соревнования
Олимпиады школьников «Шаг в будущее» по программированию, весна 2019 г.
11 класс**

Вариант 3

Задача 1

Для сокрытия своих записей некий волшебник решил использовать не общепринятую систему счисления. Однако, страдая забывчивостью, дописывал в каждом трактате и основание этой системы счисления, и обозначения элементов алфавита. Помогите чародею понять свои прошлые записи, но, чтобы не обидеть старика, дайте ответ в указанной им системе счисления.

На вход подаётся строка с основанием используемой системы счисления и перечислением элементов алфавита, а также выражение, к которому нужно указать ответ. Ответ дать в указанной системе счисления в формате строки.

Входная строка представляет собой число-основание системы счисления и последовательность разрядов, записанных через произвольное число пробелов по возрастанию значения, после чего – строковая запись выражения.

Выходной строкой является строка, содержащая ответ, который необходимо найти по заданию.

Изменять формат входных и выходных данных запрещено.

Оставлять в конце программы ожидание ввода запрещено.

Помните, что автоматическая проверка населена роботами.

Входная строка: 3 0 1 2 21+20

Результат: 111

Критерии: Простая задача – 5 баллов

N	Оценка	Входные данные	Выходные данные
1	1	4 0 1 2 3 11+10	21
2	1	5 0 1 2 3 4 31+20	101
3	1	3 0 1 2 21+20	111
4	1	7 0 1 2 3 4 5 6 51+50	131
5	1	5 a b c d e db+ca	bab

Решение:

```
#include "stdafx.h"
```

```
#include <iostream>
```

```
#include <string>
```

```
#include <cmath>
```

```
std::string convertSingleTo(int num, char *symbols)
```

```

{
    return std::string() + symbols[num];
}

std::string convertTo(int num, int base, char *symbols)
{
    if(abs(num) < base) return convertSingleTo(abs(num), symbols);
    if(num < 0)
        return "-" + (convertTo(abs(num)/base, base, symbols) + convertSingleTo(abs(num)%base,
symbols));
    return convertTo(abs(num)/base, base, symbols) + convertSingleTo(abs(num)%base, symbols);
}

int convertSingleFrom(char num, char *symbols)
{
    int i=0;
    while(symbols[i] != num) i++;
    return i;
}

int convertFrom(std::string num, int base, char *symbols)
{
    int k=1;
    if(num[0] == '-')
    {
        k = -1;
        num.erase(num.begin());
    }
    int result = 0;
    for(int i=num.length()-1; i>=0; i--)
        result += convertSingleFrom(num[i], symbols)*pow(base, num.length()-1-i);
    return k*result;
}

int findOperandPos(std::string str)
{
    char *operands = new char[5];
    operands[0] = '+';
    operands[1] = '-';
    operands[2] = '*';
    operands[3] = '/';
    operands[4] = '%';
    for(int i=0; i<str.length(); i++)
        for(int k=0; k<5; k++)
            if(str[i] == operands[k]) return i;
    return -1;
}

std::string doTask(std::string task, int base, char *symbols)
{
    for(int i=task.length(); i>=0; i--)
        if((task[i] == ')') || (task[i] == '('))

```

```

        task.erase(task.begin() + i);

int first = 1;
if(task[0] == '-')
{
    task.erase(task.begin());
    first = -1;
}

int operand = findOperandPos(task);
first *= convertFrom(task.substr(0, operand), base, symbols);

task.erase(task.begin(), task.begin() + operand);

int second = 1;
if(task[1] == '-')
{
    task.erase(task.begin() + 1);
    second = -1;
}
second *= convertFrom(task.substr(1, task.length()-1), base, symbols);

int result;
switch(task[0])
{
    case '+': result = first+second; break;
    case '-': result = first-second; break;
    case '*': result = first*second; break;
    case '/': result = first/second; break;
    case '%': result = first%second; break;
    default: result = -1; break;
}

return convertTo(result, base, symbols);
}
int main()
{
    int base;
    std::cin>>base;
    char *symbols = new char[base];
    for(int i=0; i<base; i++)
        std::cin>>symbols[i];
    std::string task;
    std::cin>>task;

    std::string result = doTask(task, base, symbols);
    std::cout<<result<<"\n";
return 0;
}

```

Задача 2

После того как ученик уронил тетрадь с примерами на пол, из тетради выпали все арифметические знаки (сложение, вычитание, умножение и целочисленное деление, а также остаток от деления). На вход программе подаётся строка с целыми числами, помогите расставить выпавшие знаки так, чтобы получилось верное равенство. Ответ запишите строкой, указывая через пробел символы арифметических операций (+, -, *, / – целочисленное деление, % – остаток от деления; операции указаны в порядке убывания частоты использования). В случае, если таких записей несколько укажите вариант с наиболее часто употребляемыми операциями.

Входная строка представляет собой последовательность чисел, записанные через произвольное число пробелов.

Выходной строкой является строка, которую необходимо найти по заданию.

Изменять формат входных и выходных данных запрещено.

Оставлять в конце программы ожидание ввода запрещено.

Помните, что автоматическая проверка населена роботами.

Входные данные: 2 9 9 9

Результат: * = +

Критерии: простая 15 баллов

N	Оценка	Входные данные	Выходные данные
1	1	5 2 14 7	+ = -
2	1	9 2 7 3 3	- - + =
3	1	2 9 9 9	* = +
4	1	3 6 10 8	* = +
5	5	1 2 3 2 17 8	+ + + % =
6	1	6 9 3	= -
7	5	81 27 9 0 6	/ - = -

Решение:

```
s=list(map(int, input().split()))
n=len(s)
slf={0:"+", 1:"- ", 2:"*", 3:"/", 4:"%", 5:"==", 6:"="}
a=[0]*(n-1)

f=0

while ( sum(a)<4*(n-1) and f==0):
    for j in range(n-1):
        q=a[j]
        a[j]=5
        ans=""
```

```

for i in range(n-1):
    ans+=str(s[i])+slf[a[i]]
ans+=str(s[-1])

if eval(ans) and f==0:
    f=1
    for k in range(n-1):
        w=a[k]
        if w==5:
            w+=1
        print(slf[w], end=" ")
    print()

```

a[j] = q

```

o=1
for i in range(n-1):
    a[i]+=o
    o=a[i]//5
    a[i]=a[i]%5

```

Задача 3

При анализе генома человека иногда есть необходимость находить повторяющиеся последовательности в строке, состоящей из алфавита А,С,Г,Т. Необходимо составить алгоритм, работающий для произвольного генома, определяющий в заданной строке наиболее длинную подстроку, которая повторяется заданное количество раз (или вернуть пустую строку).

Входная строка представляет собой число повторений и заданная строка, записанные через произвольное число пробелов.

Выходной строкой является строка, которую необходимо найти по заданию.

Изменять формат входных и выходных данных запрещено.

Оставлять в конце программы ожидание ввода запрещено.

Помните, что автоматическая проверка населена роботами.

Входная строка: 3 ACGCCGCTTTTCCGGTTCGC.

Результат: CGC

Пояснение: последовательность ТТ тоже подойдёт, но она более короткая, чем CGC.

Критерии: средняя 15 баллов

N	Оценка	Входные данные	Выходные данные
1	1	3 ACGCCGCTTTCCGGTTCGC	CGC
2	1	3 ухогорлоносоговорилородогурцов	ого
3	1	3 inhfa pf ghtdsitybt crjhjcnb cnj, inhfa pf gmzycndj ldf, inhfa pf ghjcnj nfr vyjuj	inhfa pf
4	1	2 перпендикуляр, опущенный из заданной точки К на плоскость, будет также параллелен прямой АС, которая вместе с заданной точкой К образует плоскость.	плоскость
5	5	1000 На краю дороги стоял дуб. Он был, вероятно, в десять раз старше берез, составлявших лес, в десять раз толще и в два раза выше каждой березы. Это был огромный, в два обхвата дуб, с обломанными суками и корой, заросшей старыми болячками. С огромными, неуклюже, несимметрично растопыренными корявыми руками и пальцами, он старым, сердитым и презрительным уродом стоял между улыбающимися березами.	
6	5	0	
7	1	4 ехать на конференцию на верхней полке на две недели на поезде	на

Решение:

```

var
  n,i,i2,i3,c:longint;
  s,current: string;
  checked: array[1..100000] of string;
  function rec(st: string;depth: longint; j: longint):boolean;
  begin
    if not (st[depth]=s[j]) then begin rec:= false; exit; end;
    if depth = length(st) then begin rec:= true; exit; end else rec:=rec(st,depth+1,j+1);
  end;

function count(str: string):longint;
var
  j,cou,ans: longint;
begin
  ans:=0;
  j:=1;
  for j:=1 to length(s) do
  begin
    if str[1]=s[j] then if rec(str,1,j) then ans:=ans+1;
  end;
  count:=ans;
end;

procedure check(str: string);
var
  j: longint;
begin

```

```

for j:=1 to c do
if checked[j]=str then exit;
c:=c+1;
checked[c]:=str;
if count(str)=n then begin writeln(str); halt(); end;
end;

begin
read(n);
readln(s);
c:=0;
//      n:=2;
//  s:='BCCBCCBCBCCBC';
current:="";
for i:=length(s) downto 1 do
for i2:=1 to length(s)-i do
begin
for i3:=i2 to i2+i-1 do
begin
current:=current+s[i3];
end;
check(current);
current:="";
end;
end.

```

Задача 4

Можно ли распределить группу студентов, с указанными во входной строке весами, на две примерно равные по общему весу группы? И если можно, то какими будут веса каждой группы? (ответ запишите в порядке убывания значения общего веса).

Необходимо написать консольное приложение.

Изменять формат входных и выходных данных запрещено.

Оставлять в конце программы ожидание ввода запрещено.

Помните, что автоматическая проверка населена роботами.

На вход подаётся строка с целыми числами, записанными через пробелы. Результатом является строка, состоящая из двух чисел, записанных через пробел.

Входные параметры	Выходные параметры
5 1 100 20 40 30 50	125 121
1 10	10 1

Критерии: средняя 18 баллов

N	Оценка	Входные данные	Выходные данные
1	5	5 1 100 20 40 30 50	125 121
2	5	1 10	10 1
3	8	5 10 95 75 100 115 20	210 210

Решение:

```

program Project4;
  type mas=array[1..100] of integer;
  var s2,s1:string; m:mas; i,x,y, n,c:integer; f:boolean;
  { procedure rec(s1,s2:string;x,y:integer) ;
  var a,c:integer; s3:string;
  begin
    if length(s1)=0 then
      if x=y then readln(s2);
      if length(s1)>0 then
        begin
          s3:=copy(s1,pos(' ',s1)+1,length(s1)-pos(' ',s1));
          val(copy(s1,1,pos(' ',s1)-1),a,c);
          if(length(s2)=0) then rec(s3,s2,a,y)
          else
            begin
              rec(s3,s2+"+",x+a,y);
              rec(s3,s2+"-",x-a,y);
              rec(s3,s2+"*",x*a,y);
              rec(s3,s2+"/",x/a,y);
            end;
        end;
      end;
    end;
  }
function sum(x,y:integer;m:mas):integer;
var i,s:integer;
begin
  s:=0;
  for i:= x to y do
    s:= s+m[i];
  sum:=s;
end;
begin
  readln(s1);
  n:=1;
  f:=true;
  while(pos(' ',s1)>0) do
    begin
      val(copy(s1,1,pos(' ',s1)-1),m[n],c);
      delete(s1,1,pos(' ',s1));
      n:=n+1;
    end;
  val(s1,m[n],c);

```

```

while(f) do
begin
f:=false;
for i:= 1 to n-1 do
if m[i]<m[i+1] then
begin f:= true;c:= m[i];m[i]:=m[i+1];m[i+1]:=c;end;
end;
for i:= 1 to n do
begin
if x>y then y:=y+m[i]
else x:=x+ m[i];
end;
if x>y then writeln(x,' ',y)
else writeln(y,' ',x) ;
end.

```

Задача 5

В Бородинском сражении солдаты шли в атаку в построении, сходным с прямоугольником со сторонами М и N человек. Во время движения важно «чувство локтя» в ряду, т.е. рост солдат в каждом ряду должен убывать от левого края к правому (от 0 индекса к старшему). Также важно, чтобы первый ряд состоял из наименее рослых солдат, а последний – наоборот. К сожалению, солдаты успели сдружиться рядами, поэтому перемещать солдат из одного ряда в другой нельзя. Расставьте солдат так, чтобы повысить качество их перемещения.

На вход подаётся матрица, где каждая ячейка – рост. На выходе – изменённая матрица в виде строки. Входная строка представляет собой последовательность чисел, записанные через произвольное число пробелов. Первые два числа – количество столбцов и строк.

Выходной строкой является строка с числами, записанными через пробел, которую необходимо найти по заданию.

Изменять формат входных и выходных данных запрещено.

Оставлять в конце программы ожидание ввода запрещено.

Помните, что автоматическая проверка населена роботами.

Входная строка: 3 3 9 3 5 1 2 6 2 3 4

Результат: 6 2 1 4 3 2 9 5 3

Критерии: сложная 22 баллов

N	Оценка	Входные данные	Выходные данные
1	7	6 3 9 5 8 3 7 4 6 1 2 1 9 7 4 3 6 2 5 5	6 5 5 4 3 2 9 7 6 2 1 1 9 8 7 5 4 3
2	7	3 3 9 3 5 1 2 6 2 3 4	6 2 1 4 3 2 9 5 3
3	8	0 0	

Решение:

```
l = list(map(int,input().split()))
```

```

n,m = l[0],l[1]
l = l[2:]
#n,m = map(int, input().split())
a = []
for i in range(m):
    a.append(l[:n])
    l = l[n:]
    #a.append(list(map(int,input().split())))
    a[i].sort()
    a[i].reverse()
for i in range(len(a)):
    for j in range(i,len(a)):
        if j>i and sum(a[i])>sum(a[j]):
            arr = a[i]
            a[i] = a[j]
            a[j] = arr
ans = []
for i in range(m):
    ans = ans+a[i]
print(*ans)

```

Задача 6

Планктонный рачок церкопагис (*Cercopagis pengoi*) обладает длинным хвостом (иногда длина хвоста превосходит длину тела в 7 раз). Одной из гипотез о назначении подобного хвоста, который, кроме прочего, наделён шипами, является регуляция численности: хвосты спутываются и мешают свободному перемещению для добывания пищи, что приводит к голодной смерти животного. Очевидно, что чем длиннее хвост, тем больше собратьев могут в нём запутаться. Шипы разбросаны зонами по всей длине хвоста. Считается, что планктонные рачки спутываются именно этими зонами.

На вход подаётся количество зон с шипами у каждого рачка. Определите наибольшее число церкопагисов, которые могут спутаться в одну группу.

Входная строка представляет собой последовательность целых чисел, записанных через произвольное число пробелов.

Выходной строкой является строка, содержащая число, которое необходимо найти по заданию.

Изменять формат входных и выходных данных запрещено.

Оставлять в конце программы ожидание ввода запрещено.

Помните, что автоматическая проверка населена роботами.

Входные данные: 3 2 1 1 1 1 2 1

Результат: 6

Критерии: сложная 25 баллов

N	Оценка	Входные данные	Выходные данные
---	--------	----------------	-----------------

1	1	3 1 1 1 1 1 1 1 1 2 1 1 1 1	5
2	1	3 1 1 1 1 1 1 5 1 1 1 1 1 1 1 1 1 1	8
3	1	1 2 2 2 1 1 1 1 1 1 1 1 1 1 1 4	8
4	1	3 2 1 1 1 1 1 2 1	6
5	8	9 5	2
6	7	2 2 2 2 2 2 2 8	8
7	6		0

Решение:

```

a = input().split()
rez = 0
lenth = 0
notfind = 1
for i in a:
    i = int(i)
    if notfind:
        rez += i+1
        notfind = 0
    else:
        rez += i-1
    lenth += 1
print(min(rez,lenth))

```

**Заключительный (очный) этап академического соревнования
Олимпиады школьников «Шаг в будущее» по программированию, весна 2019 г.
11 класс**

Вариант 4

Задача 1

Эльф, гном и человек шли по подземелью в поисках сокровищ. Упервшись в стену, которая заграждала проход:

- гном пришёл к выводу, что это не стена, а дверь, которую заложили мастера древности, а за дверью их ждёт сокровище;
- эльф припомнил, что мастера древности любили загадки и математические равенства;
- человек узнал алфавит используемой древними системы счисления.

После вынужденного обсуждения также были выяснены следующие факты:

- на стену нанесены числа в некой позиционной системе, причём операция находится между аргументами;
- основание системы счисления целое и положительное;
- для ответа нужно указать ответ в используемой системе счисления.

На вход подаётся строка с основанием используемой системы счисления и перечислением элементов алфавита, а также выражение, к которому нужно указать ответ. Ответ дать в указанной системе счисления в формате строки.

Входная строка представляет собой число-основание системы счисления и последовательность разрядов, записанных через произвольное число пробелов по возрастанию значения, после чего – строковая запись выражения.

Выходной строкой является строка, содержащая ответ, который необходимо найти по заданию.

Изменять формат входных и выходных данных запрещено.

Оставлять в конце программы ожидание ввода запрещено.

Помните, что автоматическая проверка населена роботами.

Входная строка: 4 0 1 2 3 11+10

Результат: 21

Критерии: Простая задача – 5 баллов

N	Оценка	Входные данные	Выходные данные
1	1	4 0 1 2 3 11+10	21
2	1	5 0 1 2 3 4 31+20	101
3	1	3 0 1 2 21+20	111

4	1	7 0 1 2 3 4 5 6 51+50	131
5	1	5 a b c d e db+ca	bab

Решение:

```
#include "stdafx.h"

#include <iostream>
#include <string>
#include <cmath>

std::string convertSingleTo(int num, char *symbols)
{
    return std::string() + symbols[num];
}

std::string convertTo(int num, int base, char *symbols)
{
    if(abs(num) < base) return convertSingleTo(abs(num), symbols);
    if(num < 0)
        return "-" + (convertTo(abs(num)/base, base, symbols) + convertSingleTo(abs(num)%base,
symbols));
    return convertTo(abs(num)/base, base, symbols) + convertSingleTo(abs(num)%base, symbols);
}

int convertSingleFrom(char num, char *symbols)
{
    int i=0;
    while(symbols[i] != num) i++;
    return i;
}

int convertFrom(std::string num, int base, char *symbols)
{
    int k=1;
    if(num[0] == '-')
    {
        k = -1;
        num.erase(num.begin());
    }
    int result = 0;
    for(int i=num.length()-1; i>=0; i--)
        result += convertSingleFrom(num[i], symbols)*pow(base, num.length()-1-i);
    return k*result;
}

int findOperandPos(std::string str)
{
    char *operands = new char[5];
    operands[0] = '+';
    operands[1] = '-';
}
```

```

operands[2] = '*';
operands[3] = '/';
operands[4] = '%';
for(int i=0; i<str.length(); i++)
    for(int k=0; k<5; k++)
        if(str[i] == operands[k]) return i;
return -1;
}

std::string doTask(std::string task, int base, char *symbols)
{
    for(int i=task.length(); i>=0; i--)
        if((task[i] == ')') || (task[i] == '('))
            task.erase(task.begin() + i);

    int first = 1;
    if(task[0] == '-')
    {
        task.erase(task.begin());
        first = -1;
    }

    int operand = findOperandPos(task);
    first *= convertFrom(task.substr(0, operand), base, symbols);

    task.erase(task.begin(), task.begin() + operand);

    int second = 1;
    if(task[1] == '-')
    {
        task.erase(task.begin() + 1);
        second = -1;
    }
    second *= convertFrom(task.substr(1, task.length()-1), base, symbols);

    int result;
    switch(task[0])
    {
        case '+': result = first+second; break;
        case '-': result = first-second; break;
        case '*': result = first*second; break;
        case '/': result = first/second; break;
        case '%': result = first%second; break;
        default: result = -1; break;
    }

    return convertTo(result, base, symbols);
}

int main()
{
    int base;

```

```

std::cin>>base;
char *symbols = new char[base];
for(int i=0; i<base; i++)
    std::cin>>symbols[i];
std::string task;
std::cin>>task;

std::string result = doTask(task, base, symbols);
std::cout<<result<<"\n";
return 0;
}

```

Задача 2

Марья Ивановна решила проверить работы по ЕГЭ, но весенний паводок смыл остатки знаний, которые учащиеся оставили на бланках. Решив помочь отрокам, Марья Ивановна приступила к расстановке смытых арифметических знаков (сложение, вычитание, умножение и целочисленное деление, а также остаток от деления). На вход программе подаётся строка с целыми числами, которые были напечатаны несмываемой краской (в отличии от чернил ручек), помогите Марье Ивановне расставить знаки так, чтобы получилось верное равенство. Ответ запишите строкой, указывая через пробел символы арифметических операций (+, -, *, / – целочисленное деление, % – остаток от деления; операции указаны в порядке убывания частоты использования). В случае, если таких записей несколько укажите вариант с наиболее часто употребляемыми операциями.

Входная строка представляет собой последовательность чисел, записанные через произвольное число пробелов.

Выходной строкой является строка, которую необходимо найти по заданию.

Изменять формат входных и выходных данных запрещено.

Оставлять в конце программы ожидание ввода запрещено.

Помните, что автоматическая проверка населена роботами.

Входные данные: 9 2 7 3 3

Результат: - - + =

Критерии: простая 15 баллов

N	Оценка	Входные данные	Выходные данные
1	1	5 2 14 7	+ = -
2	1	9 2 7 3 3	- - + =
3	1	2 9 9 9	* = +
4	1	3 6 10 8	* = +
5	5	1 2 3 2 17 8	+ + + % =
6	1	6 9 3	= -
7	5	81 27 9 0 6	/ - = -

Решение:

```
s=list(map(int, input().split()))
n=len(s)
slf={0:"+", 1:"- ", 2:"*", 3:"/", 4:"%", 5:"==", 6:"="}
a=[0]*(n-1)
```

```
f=0
```

```
while ( sum(a)<4*(n-1) and f==0):
```

```
    for j in range(n-1):
```

```
        q=a[j]
```

```
        a[j]=5
```

```
        ans=""
```

```
        for i in range(n-1):
```

```
            ans+=str(s[i])+slf[a[i]]
```

```
        ans+=str(s[-1])
```

```
    if eval(ans) and f==0:
```

```
        f=1
```

```
        for k in range(n-1):
```

```
            w=a[k]
```

```
            if w==5:
```

```
                w+=1
```

```
            print(slf[w], end=" ")
```

```
        print()
```

```
    a[j] = q
```

```
o=1
```

```
for i in range(n-1):
```

```
    a[i]+=o
```

```
    o=a[i]//5
```

```
    a[i]=a[i]%5
```

Задача 3

При чтении текста Вася и Петя решили удалить в каждой строке пробелы и загадать число, чтобы найти наиболее длинную последовательность слов, которая повторяется в этой строке загаданное число раз. Также рассмотрите случай, когда пробелы не удаляются.

Входная строка представляет собой число повторений и заданная строка, записанные через произвольное число пробелов.

Выходной строкой является строка, которую необходимо найти по заданию.

Изменять формат входных и выходных данных запрещено.

Оставлять в конце программы ожидание ввода запрещено.

Помните, что автоматическая проверка населена роботами.

Входная строка: 3 ухогорлоносоговорилородогурцов

Результат: ого

Пояснение: 3 – заданное количество повторений, после него идёт строка.

Критерии: средняя 15 баллов

N	Оценка	Входные данные	Выходные данные
1	1	3 ACGCCGCTTTCCGGTTCGC	CGC
2	1	3 ухогорлоносоговорилородогурцов	ого
3	1	3 inhfa pf ghtdsitybt crjhcnb cnj, inhfa pf gmzycndj ldf, inhfa pf ghjcnj nfr vyujj	inhfa pf
4	1	2 перпендикуляр, опущенный из заданной точки К на плоскость, будет также параллелен прямой АС, которая вместе с заданной точкой К образует плоскость.	плоскость
5	5	1000 На краю дороги стоял дуб. Он был, вероятно, в десять раз старше берез, составлявших лес, в десять раз толще и в два раза выше каждой березы. Это был огромный, в два обхвата дуб, с обломанными суками и корой, заросшей старыми болячками. С огромными, неуклюже, несимметрично растопыренными корявыми руками и пальцами, он старым, сердитым и презрительным уродом стоял между улыбающимися березами.	
6	5	0	
7	1	4 ехать на конференцию на верхней полке на две недели на поезде	на

Решение:

```
var
  n,i,i2,i3,c:longint;
  s,current: string;
  checked: array[1..100000] of string;
  function rec(st: string;depth: longint; j: longint):boolean;
  begin
    if not (st[depth]=s[j]) then begin rec:= false; exit; end;
    if depth = length(st) then begin rec:= true; exit; end else rec:=rec(st,depth+1,j+1);
  end;

function count(str: string):longint;
var
  j,cou,ans: longint;
begin
  ans:=0;
  j:=1;
  for j:=1 to length(s) do
  begin
    if str[1]=s[j] then if rec(str,1,j) then ans:=ans+1;
  end;
  count:=ans;
```

```

end;

procedure check(str: string);
var
  j: longint;
begin
  for j:=1 to c do
    if checked[j]=str then exit;
    c:=c+1;
    checked[c]:=str;
    if count(str)=n then begin writeln(str); halt(); end;
  end;

begin
read(n);
readln(s);
  c:=0;
//      n:=2;
//  s:='BCCBBCCBCBCCBC';
  current:="";
  for i:=length(s) downto 1 do
    for i2:=1 to length(s)-i do
      begin
        for i3:=i2 to i2+i-1 do
          begin
            current:=current+s[i3];
          end;
          check(current);
          current:="";
        end;
      end;
    end.

```

Задача 4

Вася и Коля поспорили, что если особым образом связать шнурки от ботинок у всего класса, то можно получить две верёвки одинаковой длины. Однако Вася не учёл, что не у всех присутствующих шнурки одинаковой длины. Укажите длины верёвок, которые можно получить при связывании, так, чтобы их длины были примерно одинаковы (размером узлов пренебречь).

На вход подаётся строка с длинами шнурков (только целые числа), записанных через пробел. На выходе ожидается строка с двумя длинами, записанными через пробел в порядке убывания длины.

Изменять формат входных и выходных данных запрещено.

Оставлять в конце программы ожидание ввода запрещено.

Помните, что автоматическая проверка населена роботами.

Входные параметры	Выходные параметры
5 1 100 20 40 30 50	125 121
1 10	10 1

Критерии: средняя 18 баллов

N	Оценка	Входные данные	Выходные данные
1	5	5 1 100 20 40 30 50	125 121
2	5	1 10	10 1
3	8	5 10 95 75 100 115 20	210 210

Решение:

```

program Project4;
  type mas=array[1..100] of integer;
  var s2,s1:string; m:mas; i,x,y, n,c:integer; f:boolean;
  { procedure rec(s1,s2:string;x,y:integer) ;
  var a,c:integer; s3:string;
  begin
    if length(s1)=0 then
    if x=y then readln(s2);
    if length(s1)>0 then
    begin
      s3:=copy(s1,pos(' ',s1)+1,length(s1)-pos(' ',s1));
      val(copy(s1,1,pos(' ',s1)-1),a,c);
      if(length(s2)=0) then rec(s3,s2,a,y)
      else
      begin
        rec(s3,s2+"+",x+a,y);
        rec(s3,s2+"-",x-a,y);
        rec(s3,s2+"*",x*a,y);
        rec(s3,s2+"/",x/a,y);
      end;
    end;
  end;
  }
function sum(x,y:integer;m:mas):integer;
var i,s:integer;
begin
  s:=0;
  for i:= x to y do
  s:= s+m[i];
  sum:=s;
end;
begin
  readln(s1);
  n:=1;

```

```

f:=true;
while(pos(' ',s1)>0) do
begin
val(copy(s1,1,pos(' ',s1)-1),m[n],c);
delete(s1,1,pos(' ',s1));
n:=n+1;
end;
val(s1,m[n],c);

while(f) do
begin
f:=false;
for i:= 1 to n-1 do
if m[i]<m[i+1] then
begin f:= true;c:= m[i];m[i]:=m[i+1];m[i+1]:=c;end;
end;
for i:= 1 to n do
begin
if x>y then y:=y+m[i]
else x:=x+ m[i];
end;
if x>y then writeln(x,' ',y)
else writeln(y,' ',x) ;
end.

```

Задача 5

Римские манипулы состояли из людей, построенных в виде прямоугольника со сторонами М и N. Для повышения боеспособности манипулы заполнялись по увеличению опыта ряда (чем меньше опыта у членов ряда, тем ближе этот ряд к передовой), а распределение внутри ряда – по уменьшению опыта слева направо (на 0й позиции стоит человек с наибольшим опытом, далее по не возрастанию до крайней позиции). В легион прибывают отшагавшие новобранцы, построившиеся манипулой. Центурион не хочет понижать боевой дух, разделяя успешных сдружиться членов ряда. Помогите центуриону расположить легионеров, согласно требованиям.

На вход подаётся матрица, где каждая ячейка – имеющийся опыт. На выходе – изменённая матрица в виде строки.

Входная строка представляет собой последовательность чисел, записанные через произвольное число пробелов. Первые два числа – количество столбцов и строк.

Выходной строкой является строка с числами, записанными через пробел, которую необходимо найти по заданию.

Изменять формат входных и выходных данных запрещено.

Оставлять в конце программы ожидание ввода запрещено.

Помните, что автоматическая проверка населена роботами.

Входная строка: 3 3 9 3 5 1 2 6 2 3 4

Результат: 6 2 1 4 3 2 9 5 3

Критерии: сложная 22 баллов

N	Оценка	Входные данные	Выходные данные
1	7	6 3 9 5 8 3 7 4 6 1 2 1 9 7 4 3 6 2 5 5	6 5 5 4 3 2 9 7 6 2 1 1 9 8 7 5 4 3
2	7	3 3 9 3 5 1 2 6 2 3 4	6 2 1 4 3 2 9 5 3
3	8	0 0	

Решение:

```
l = list(map(int,input().split()))

n,m = l[0],l[1]
l = l[2:]
#n,m = map(int, input().split())
a = []
for i in range(m):
    a.append(l[:n])
    l = l[n:]
    #a.append(list(map(int,input().split())))
    a[i].sort()
    a[i].reverse()
for i in range(len(a)):
    for j in range(i,len(a)):
        if j>i and sum(a[i])>sum(a[j]):
            arr = a[i]
            a[i] = a[j]
            a[j] = arr
ans = []
for i in range(m):
    ans = ans+a[i]
print(*ans)
```

Задача 6

Погружаясь на большую глубину, исследователи морских осьминогов заметили своеобразные «хороводы», в которых участвует большое количество осьминогов. Каждый осьминог хватается другого осьминога за щупальце (или несколько щупальцев), но соединение 1 к 1 (одно щупальце хватателя к одному щупальцу хватаемого), причём «хоровод» получается не обязательно замкнутый.

На вход подаётся количество щупалец у каждого осьминога. Укажите, какое наибольшее количество осьминогов будет участвовать в самом большом «хороводе».

Входная строка представляет собой последовательность целых чисел, записанных через произвольное число пробелов.

Выходной строкой является строка, содержащая число, которое необходимо найти по заданию.

Изменять формат входных и выходных данных запрещено.

Оставлять в конце программы ожидание ввода запрещено.

Помните, что автоматическая проверка населена роботами.

Вход: 3 1 1 1 1 1 1 5 1 1 1 1 1 1 1 1 1 1

Выход: 8

Критерии: сложная 25 баллов

N	Оценка	Входные данные	Выходные данные
1	1	3 1 1 1 1 1 1 1 2 1 1 1 1	5
2	1	3 1 1 1 1 1 1 5 1 1 1 1 1 1 1 1 1 1	8
3	1	1 2 2 2 1 1 1 1 1 1 1 1 1 1 1 4	8
4	1	3 2 1 1 1 1 1 2 1	6
5	8	9 5	2
6	7	2 2 2 2 2 2 8	8
7	6		0

Решение:

```
a = input().split()
rez = 0
lenth = 0
notfind = 1
for i in a:
    i = int(i)
    if notfind:
        rez += i+1
        notfind = 0
    else:
        rez += i-1
    lenth += 1
print(min(rez,lenth))
```

**Заключительный (очный) этап академического соревнования
Олимпиады школьников «Шаг в будущее» по программированию, весна 2019 г.
11 класс**

Вариант 5

Задача 1

Сервер олимпиады по информатике был поработан ИИ (искусственным интеллектом), написанным на функциональном языке программирования. Чтобы запутать школьников, ИИ использует произвольную систему счисления и задаёт примеры на арифметику в ней, требуя ответ именно в указанной системе счисления. В качестве подсказки указывает основание этой системы счисления и используемый алфавит. Помогите школьникам написать олимпиаду по информатике (можно начать с себя).

На вход подаётся строка с основанием используемой системы счисления и перечислением элементов алфавита, а также выражение, к которому нужно указать ответ. Ответ дать в указанной системе счисления в формате строки.

Входная строка представляет собой число-основание системы счисления и последовательность разрядов, записанных через произвольное число пробелов по возрастанию значения, после чего – строковая запись выражения.

Выходной строкой является строка, содержащая ответ, который необходимо найти по заданию.

Изменять формат входных и выходных данных запрещено.

Оставлять в конце программы ожидание ввода запрещено.

Помните, что автоматическая проверка населена роботами.

Входная строка: 7 0 1 2 3 4 5 6 51+50

Результат: 131

Критерии: Простая задача – 5 баллов

N	Оценка	Входные данные	Выходные данные
1	1	4 0 1 2 3 11+10	21
2	1	5 0 1 2 3 4 31+20	101
3	1	3 0 1 2 21+20	111
4	1	7 0 1 2 3 4 5 6 51+50	131
5	1	5 a b c d e db+ca	bab

Решение:

```

#include "stdafx.h"

#include <iostream>
#include <string>
#include <cmath>

std::string convertSingleTo(int num, char *symbols)
{
    return std::string() + symbols[num];
}

std::string convertTo(int num, int base, char *symbols)
{
    if(abs(num) < base) return convertSingleTo(abs(num), symbols);
    if(num < 0)
        return "-" + (convertTo(abs(num)/base, base, symbols) + convertSingleTo(abs(num)%base,
symbols));
    return convertTo(abs(num)/base, base, symbols) + convertSingleTo(abs(num)%base, symbols);
}

int convertSingleFrom(char num, char *symbols)
{
    int i=0;
    while(symbols[i] != num) i++;
    return i;
}

int convertFrom(std::string num, int base, char *symbols)
{
    int k=1;
    if(num[0] == '-')
    {
        k = -1;
        num.erase(num.begin());
    }
    int result = 0;
    for(int i=num.length()-1; i>=0; i--)
        result += convertSingleFrom(num[i], symbols)*pow(base, num.length()-1-i);
    return k*result;
}

int findOperandPos(std::string str)
{
    char *operands = new char[5];
    operands[0] = '+';
    operands[1] = '-';
    operands[2] = '*';
    operands[3] = '/';
    operands[4] = '%';
    for(int i=0; i<str.length(); i++)
        for(int k=0; k<5; k++)
            if(str[i] == operands[k]) return i;
}

```

```

        return -1;
    }

std::string doTask(std::string task, int base, char *symbols)
{
    for(int i=task.length(); i>=0; i--)
        if((task[i] == ')') || (task[i] == '('))
            task.erase(task.begin() + i);

    int first = 1;
    if(task[0] == '-')
    {
        task.erase(task.begin());
        first = -1;
    }

    int operand = findOperandPos(task);
    first *= convertFrom(task.substr(0, operand), base, symbols);

    task.erase(task.begin(), task.begin() + operand);

    int second = 1;
    if(task[1] == '-')
    {
        task.erase(task.begin() + 1);
        second = -1;
    }
    second *= convertFrom(task.substr(1, task.length()-1), base, symbols);

    int result;
    switch(task[0])
    {
        case '+': result = first+second; break;
        case '-': result = first-second; break;
        case '*': result = first*second; break;
        case '/': result = first/second; break;
        case '%': result = first%second; break;
        default: result = -1; break;
    }

    return convertTo(result, base, symbols);
}

int main()
{
    int base;
    std::cin>>base;
    char *symbols = new char[base];
    for(int i=0; i<base; i++)
        std::cin>>symbols[i];
    std::string task;
    std::cin>>task;
}

```

```
    std::string result = doTask(task, base, symbols);
    std::cout<<result<<"\n";
return 0;
}
```

Задача 2

Эльф, гном и человек шли по подземелью в поисках сокровищ. Уперевшись в стену, которая заграждала проход:

- гном пришёл к выводу, что это не стена, а дверь, которую заложили мастера древности, а за дверью их ждёт сокровище;
- человек нашёл на полу невообразимое количество фиолетовых, синих, жёлтых и красных камешков, но лишь один зелёный;
- эльф припомнил, что мастера древности любили загадки и математические равенства;

После вынужденного обсуждения также были выяснены следующие факты:

- на стену нанесены барельефы чисел;
- между числами находятся пазы (куда успешно помещаются найденные разноцветные камешки – по одному в паз);
- мастера древности рассматривали цвета как знаки арифметических операций:
 - сложение – фиолетовым;
 - умножение – красным;
 - вычитание – синим;
 - равенство – зелёным;
 - целочисленное деление – жёлтым;
 - остаток от деления – оранжевым;
- древняя культура использовала только целые числа.

На вход программе подаётся строка с целыми числами, нанесённых на стену, расставьте камни так, чтобы получилось математическое равенство. Ответ запишите строкой, указывая через пробел символы (+, -, *, / – целочисленное деление, % – остаток от деления; операции указаны в порядке убывания частоты использования) арифметических операций, которые символизируются нужными цветами. В случае, если таких записей несколько укажите вариант с наиболее часто употребляемыми операциями.

Входная строка представляет собой последовательность чисел, записанные через произвольное число пробелов.

Выходной строкой является строка, которую необходимо найти по заданию.

Изменять формат входных и выходных данных запрещено.

Оставлять в конце программы ожидание ввода запрещено.

Помните, что автоматическая проверка населена роботами.

Входные данные: 5 2 14 7

Результат: + = -

Критерии: простая 15 баллов

N	Оценка	Входные данные	Выходные данные
1	1	5 2 14 7	+ = -
2	1	9 2 7 3 3	- - + =
3	1	2 9 9 9	* = +
4	1	3 6 10 8	* = +
5	5	1 2 3 2 17 8	+ + + % =
6	1	6 9 3	= -
7	5	81 27 9 0 6	/ - = -

Решение:

```
s=list(map(int, input().split()))
n=len(s)
slf={0:"+", 1:"- ", 2:"*", 3:"/", 4:"% ", 5:"==", 6:"="}
a=[0]*(n-1)

f=0

while ( sum(a)<4*(n-1) and f==0):
    for j in range(n-1):
        q=a[j]
        a[j]=5
        ans=""
        for i in range(n-1):
            ans+=str(s[i])+slf[a[i]]
        ans+=str(s[-1])

        if eval(ans) and f==0:
            f=1
            for k in range(n-1):
                w=a[k]
                if w==5:
                    w+=1
                print(slf[w], end=" ")
            print()

        a[j] = q

o=1
for i in range(n-1):
```

```

a[i]+=o
o=a[i]/5
a[i]=a[i]%5

```

Задача 3

Какое слово повторяется в учебнике по начертательной геометрии наибольшее число раз? (в методичке это было слово «точка»). А какое самое длинное слово там встречается чаще других? (перпендикулярные). А какая последовательность букв будет встречаться в тексте N раз? Напишите программу, которая позволит найти наиболее длинную последовательность слов, которая повторяется в заданной строке выбранное число раз и узнаете!

Входная строка представляет собой число повторений и заданная строка, записанные через произвольное число пробелов.

Выходной строкой является строка, которую необходимо найти по заданию.

Изменять формат входных и выходных данных запрещено.

Оставлять в конце программы ожидание ввода запрещено.

Помните, что автоматическая проверка населена роботами.

Входная строка: 2 перпендикуляр, опущенный из заданной точки K на плоскость, будет также параллелен прямой AC, которая вместе с заданной точкой K образует плоскость.

Результат: «плоскость» *прим.авторов* (кавычек нет)

Пояснение: 2 – заданное количество повторений, после него идёт строка.

Критерии: средняя 15 баллов

N	Оценка	Входные данные	Выходные данные
1	1	3 ACGCCGCTTTCCGGTTCGC	CGC
2	1	3 ухогорлоносоговорилородогурцов	ого
3	1	3 inhfa pf ghtdsitybt crjhjcnb cnj, inhfa pf gmzycndj ldf, inhfa pf ghjcnj nfr vyuj	inhfa pf
4	1	2 перпендикуляр, опущенный из заданной точки K на плоскость, будет также параллелен прямой AC, которая вместе с заданной точкой K образует плоскость.	плоскость
5	5	1000 На краю дороги стоял дуб. Он был, вероятно, в десять раз старше берез, составлявших лес, в десять раз толще и в два раза выше каждой березы. Это был огромный, в два обхвата дуб, с обломанными суками и корой, заросшей старыми болячками. С огромными, неуклюже, несимметрично растопыренными корявыми руками и пальцами, он старым, сердитым и презрительным уродом стоял между улыбающимися березами.	
6	5	0	
7	1	4 ехать на конференцию на верхней полке на две недели на поезде	на

Решение:

```

var
  n,i,i2,i3,c:longint;
  s,current: string;
  checked: array[1..100000] of string;
function rec(st: string;depth: longint; j: longint):boolean;
begin
  if not (st[depth]=s[j]) then begin rec:= false; exit; end;
  if depth = length(st) then begin rec:= true; exit; end else rec:=rec(st,depth+1,j+1);

end;

function count(str: string):longint;
var
  j,cou,ans: longint;
begin
  ans:=0;
  j:=1;
  for j:=1 to length(s) do
  begin
    if str[1]=s[j] then if rec(str,1,j) then ans:=ans+1;
  end;
  count:=ans;
end;

procedure check(str: string);
var
  j: longint;
begin
  for j:=1 to c do
  if checked[j]=str then exit;
  c:=c+1;
  checked[c]:=str;
  if count(str)=n then begin writeln(str); halt(); end;
end;

begin
read(n);
readln(s);
  c:=0;
//      n:=2;
//  s:='BCCBBCCBCBBCCBC';
  current:="";
  for i:=length(s) downto 1 do
  for i2:=1 to length(s)-i do
  begin
    for i3:=i2 to i2+i-1 do
    begin
      current:=current+s[i3];
    end;
    check(current);
    current:="";
  end;
end;

```

end.

Задача 4

Можно купить 2 тягача по цене одного для транспортировки грузов. Определите тяговую способность одного тягача, чтобы нагрузка была примерно одинакова для обоих тягачей.

На вход подаётся строка с весами грузов, которые нужно доставить по железной дороге, записанных через пробел.

На выходе ожидается строка с двумя числами, записанными через пробел в порядке убывания, соответствующие тяговой нагрузке на каждый из сформированных составов.

Изменять формат входных и выходных данных запрещено.

Оставлять в конце программы ожидание ввода запрещено.

Помните, что автоматическая проверка населена роботами.

Входные параметры	Выходные параметры
5 1 100 20 40 30 50	125 121
1 10	10 1

Критерии: средняя 18 баллов

N	Оценка	Входные данные	Выходные данные
1	5	5 1 100 20 40 30 50	125 121
2	5	1 10	10 1
3	8	5 10 95 75 100 115 20	210 210

Решение:

```
program Project4;
  type mas=array[1..100] of integer;
  var s2,s1:string; m:mas; i,x,y, n,c:integer; f:boolean;
  { procedure rec(s1,s2:string;x,y:integer) ;
  var a,c:integer; s3:string;
  begin
    if length(s1)=0 then
    if x=y then readln(s2);
    if length(s1)>0 then
    begin
      s3:=copy(s1,pos(' ',s1)+1,length(s1)-pos(' ',s1));
      val(copy(s1,1,pos(' ',s1)-1),a,c);
      if(length(s2)=0) then rec(s3,s2,a,y)
      else
      begin
        rec(s3,s2+"+",x+a,y);
        rec(s3,s2+"-",x-a,y);
```

```

        rec(s3,s2+"*",x*a,y);
        rec(s3,s2+"/",x/a,y);
    end;

    end;
end;
}
function sum(x,y:integer;m:mas):integer;
var i,s:integer;
begin
    s:=0;
    for i:= x to y do
        s:= s+m[i];
        sum:=s;
    end;
begin
    readln(s1);
    n:=1;
    f:=true;
    while(pos(' ',s1)>0) do
        begin
            val(copy(s1,1,pos(' ',s1)-1),m[n],c);
            delete(s1,1,pos(' ',s1));
            n:=n+1;
        end;
        val(s1,m[n],c);

    while(f) do
        begin
            f:=false;
            for i:= 1 to n-1 do
                if m[i]<m[i+1] then
                    begin f:= true;c:= m[i];m[i]:=m[i+1];m[i+1]:=c;end;
                end;
            for i:= 1 to n do
                begin
                    if x>y then y:=y+m[i]
                    else x:=x+ m[i];
                end;
            if x>y then writeln(x,' ',y)
            else writeln(y,' ',x) ;
        end.

```

Задача 5

В классе размещаются N рядов по M парт в каждом. За каждой партой сидит 1 школьник. Необходимо рассадить детей так, чтобы выполнялись условия:

- отдельно школьников можно перемещать только между рядами;
- самые высокие школьники должны сидеть на левом (нулевом) ряду, а самые низкие – на правом;
- парты, равноудалённые от доски, должны быть заняты с повышением среднего роста сидящих там школьников;

На вход подаётся матрица, где каждая ячейка – рост школьника. На выходе – изменённая матрица. Входная строка представляет собой последовательность чисел, записанные через произвольное число пробелов. Первые два числа – количество столбцов и строк.

Выходной строкой является строка с числами, записанными через пробел, которую необходимо найти по заданию.

Изменять формат входных и выходных данных запрещено.

Оставлять в конце программы ожидание ввода запрещено.

Помните, что автоматическая проверка населена роботами.

Входная строка: 6 3 9 5 8 3 7 4 6 1 2 1 9 7 4 3 6 2 5 5

Выходная строка 6 5 5 4 3 2 9 7 6 2 1 1 9 8 7 5 4 3

Критерии: сложная 22 баллов

N	Оценка	Входные данные	Выходные данные
1	7	6 3 9 5 8 3 7 4 6 1 2 1 9 7 4 3 6 2 5 5	6 5 5 4 3 2 9 7 6 2 1 1 9 8 7 5 4 3
2	7	3 3 9 3 5 1 2 6 2 3 4	6 2 1 4 3 2 9 5 3
3	8	0 0	

Решение:

```
l = list(map(int,input().split()))
```

```
n,m = l[0],l[1]
```

```
l = l[2:]
```

```
#n,m = map(int, input().split())
```

```
a = []
```

```
for i in range(m):
```

```
    a.append(l[:n])
```

```
    l = l[n:]
```

```
    #a.append(list(map(int,input().split())))
```

```
    a[i].sort()
```

```
    a[i].reverse()
```

```
for i in range(len(a)):
```

```
    for j in range(i,len(a)):
```

```
        if j>i and sum(a[i])>sum(a[j]):
```

```

arr = a[i]
a[i] = a[j]
a[j] = arr
ans = []
for i in range(m):
    ans = ans+a[i]
print(*ans)

```

Задача 6

Леший борется с незаконной вырубкой леса: он спутывает ветви деревьев так, что дерево, которое срубают, держится кроной за соседние деревья и не падает. Таким образом, чтобы свалить одно дерево нужно срубить все деревья за которые держится первое дерево. Также, пока никто не видит, леший может переставлять деревья поближе друг к другу.

На вход подаётся строка с количеством ветвей, способных к соединению, укажите какое наибольшее число деревьев может быть соединено в единую структуру?

Входная строка представляет собой последовательность целых чисел, записанных через произвольное число пробелов.

Выходной строкой является строка, содержащая число, которое необходимо найти по заданию.

Изменять формат входных и выходных данных запрещено.

Оставлять в конце программы ожидание ввода запрещено.

Помните, что автоматическая проверка населена роботами.

Вход: 3 1 1 1 1 1 1 1 1 2 1 1 1 1

Выход: 5

Критерии: сложная 25 баллов

N	Оценка	Входные данные	Выходные данные
1	1	3 1 1 1 1 1 1 1 1 2 1 1 1 1	5
2	1	3 1 1 1 1 1 1 5 1 1 1 1 1 1 1 1 1 1	8
3	1	1 2 2 2 1 1 1 1 1 1 1 1 1 1 1 4	8
4	1	3 2 1 1 1 1 1 2 1	6
5	8	9 5	2
6	7	2 2 2 2 2 2 2 8	8
7	6		0

Решение:

```

a = input().split()
rez = 0
lenth = 0

```

```
notfind = 1
for i in a:
    i = int(i)
    if notfind:
        rez += i+1
        notfind = 0
    else:
        rez += i-1
    lenth += 1
print(min(rez,lenth))
```