

**Заключительный (очный) этап научно-образовательного соревнования
Олимпиады школьников «Шаг в будущее» по профилю «Инженерное дело» специализации
«Техника и технологии» (общеобразовательный предмет информатика), весна 2019 г.**

11 класс

Вариант 1

Распределение баллов по заданиям:

Номер задачи	1	2	3	4
Баллы	15	15	18	22

Задание 1

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: + = -

Тесты

№	Вес теста	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
```

Задание 2

Чтобы запутать Шерлока Холмса Мориарти уговорил знакомого писать доктору Ватсону письма, составленные из слов, вырезанных из выпусков «Дэйли Телеграф», но если вырезанное слово состоит из количества символов, совпадающего с простым числом Фибоначчи, то рядом с ним нужно вклеить такое же слово из свежего выпуска «Гардиан». Напишите программу, принимающую на вход строку (длинной менее 256 символов), которую хотят записать газетными вырезками, и возвращающую строку, которую будет наблюдать Ватсон.

Необходимо написать консольное приложение.

Изменять формат входных и выходных данных запрещено.

Оставлять в конце программы ожидание ввода запрещено.

Помните, что автоматическая проверка населена роботами.

Примечание: последовательность Фибоначчи — это последовательность, где первые два числа равны 0 и 1, соответственно, а каждое последующее число равно сумме двух предыдущих чисел.

№	Вес теста	Входные данные	Выходные данные
1	2	Как готовить важно	Как Как готовить важно важно
2	3		


```

end;
if k=0 then s2:=s2+s1+' ';
delete(s,1,q);
q:=pos(' ',s);
end;
writeln(s2);
end.

```

Задание 3

Вводится строка (длиной менее 256 символов), необходимо вернуть строку без палиндромов (те, что читаются слева направо также, как и справа налево, например «мама» не является палиндромом, а «лол» является), заодно удалить те слова, из букв которых можно составить слова-палиндромы, сохраняя пробелы.

Необходимо написать консольное приложение.

Изменять формат входных и выходных данных запрещено.

Оставлять в конце программы ожидание ввода запрещено.

Тесты

№	Вес теста	Входные данные	Выходные данные
1	1	ололо ололо я водитель НЛЮ	водитель НЛЮ
2	1		
3	1	Ололо ололо из вговорг	из
4	1	Чтоб исходник кремерк и кермерк	Чтоб исходник
5	5	ччттоо ссллуучччииллооссьь с тттообббооойй	тттообббооойй

Решение

```

def check(base, temp):
    ans = -1;
    for i in range(len(base)):
        if (base[i] == temp):
            ans = i;
            return (ans);
            break;
    if (ans == -1):
        return ans;

def test(s):
    base_letters = [];
    base_num_letters = [];
    for i in range(len(s)):
        temp = check(base_letters, s[i]);

```

```

if (temp != -1):
    base_num_letters[temp] += 1;
else:
    base_letters.append(s[i]);
    base_num_letters.append(1);
ans = 0;
for i in range(len(base_num_letters)):
    if (base_num_letters[i] % 2 == 1):
        ans += 1;
if (ans > 1):
    return 0; #нельзя сделать париндром
else:
    return 1; #можно сделать палиндром

def test_palindrom(s):
if (len(s) % 2 == 0):
    first = s[0:len(s) // 2]
    second = s[len(s) // 2:];
    ans = 0;
    for i in range(len(first)):
        if (first[i] != second[len(second) - i - 1]):
            ans = -1;
            break;
    if (ans == -1):
        return 0;
    else:
        return 1;
else:
    first = s[0:len(s) // 2]
    second = s[len(s) // 2 + 1:];
    ans = 0;
    for i in range(len(first)):
        if (first[i] != second[len(second) - i - 1]):
            ans = -1;
            break;
    if (ans == -1):
        return 0;
    else:
        return 1;
s = input();

def main(s):
s1 = (s).split();
ans = "";
for i in range(len(s1)):
    if (test_palindrom(s1[i]) == 1):
        ans += " ";
    elif (test(s1[i]) == 1):
        ans += " ";
    else:
        ans += s1[i] + " ";
print(ans[:-1])
main(s);

```

Задание 4

Римские манипулы состояли из людей, построенных в виде прямоугольника со сторонами M 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 сдружившихся 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

Тесты

№ теста	Вес теста	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)
```