

Решение типового варианта для 11 класса

Распределение баллов по заданиям:

Номер задачи	1	2	3	4
Баллы	15	15	18	22

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_permutatons(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=" ")

```

**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.


```

while f1<=length(s1) do
begin
d:=0;
for j:=1 to f1 do begin
if f1 mod j =0 then d:=d+1; end;
if (length(s1)=f1) and (d<=2) then
begin
k:=1;
s2:=s2+s1+' '+s1+' ';
break;
end;
p:=f1;
f1:=f1+f0;
f0:=p;
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 НЛО	водитель НЛО
2	1		
3	1	Ололо ололо из вговорг	из
4	1	Чтоб исходник кремерк и кермерк	Чтоб исходник

5	5	ччттоо ссллүүчччииллооссьь с тттоообббоооййй	тттоообббоооййй
---	---	--	-----------------

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 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

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)
```

**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 2

Распределение баллов по заданиям:

Номер задачи	1	2	3	4
Баллы	15	15	18	22

Задание 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;
- 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: + = -

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):
```


			<pre> FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF rr rr </pre>
4	5	<pre> Что ты сделал для науки eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee eeeeeeeeeeeeeeee </pre>	<pre> Что Что ты ты сделал для для науки науки eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee eee eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee eee </pre>
5	3	<pre> А потом она сказала аааааааааааа !! </pre>	<pre> А потом потом она она сказала аааааааааааа аааааааааааа !! !! </pre>

Решение

```

var q,i,k,p,f1,f0,j,d:integer;
    s, s1, s2:string;
begin
  readln(s);

  s:=s+' ';
  q:=pos(' ',s);
  s2:="";
  while q>0 do
  begin
    k:=0;
    f0:=1;
    f1:=1;
    s1:=copy(s,1,q-1);
    while f1<=length(s1) do
    begin
      d:=0;
      for j:=1 to f1 do begin
        if f1 mod j =0 then d:=d+1; end;
      if (length(s1)=f1) and (d<=2) then
        begin
          k:=1;
          s2:=s2+s1+' '+s1+' ';
          break;
        end;
      p:=f1;
      f1:=f1+f0;
      f0:=p;
    end;
  end;
end;

```

```

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

```

Задание 3

Модератор на форуме избавляется от всех запрещённых слов (набор символов между пробелами), сохраняя пробелы. Запрещёнными словами являются слова-палиндромы (те, что читаются слева направо также, как и справа налево, например «мама» не является палиндромом, а «лол» является). Заодно удаляются те слова, из букв которых можно составить запрещённые слова.

Вводится строка (длиной менее 256 символов), необходимо вернуть строку без запрещённых слов, сохраняя пробелы.

Необходимо написать консольное приложение.

Изменять формат входных и выходных данных запрещено.

Оставлять в конце программы ожидание ввода запрещено.

Помните, что автоматическая проверка населена роботами.

Тесты

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

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 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

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)
```

**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 3

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 поздравить Леонардо Пизанского с днём рождения, его коллеги решили написать поздравление особым образом: если встречается слово, состоящее из количества символов, совпадающего с простым числом Фибоначчи, то его нужно записать ещё раз. Напишите программу, принимающую на вход строку (длинной менее 256 символов), которая содержит неизменённый текст, и возвращающую строку, которая будет записана в поздравлении.

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		
3	5	FFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFF r r	FFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFF r r
4	5	Что ты сделал для науки eeeeeeeeeeeeeeeeeeeeeeeeeee eeeeeeeeeeeeeeeeeeeeeeeeeee eeeeeeeeeeeeeeeeeeeeeeeeeee	Что Что ты ты сделал для для науки науки eeeeeeeeeeeeeeeeeeeeeeeeeee eeeeeeeeeeeeeeeeeeeeeeeeeee eee eeeeeeeeeeeeeeeeeeeeeeeeeee eeeeeeeeeeeeeeeeeeeeeeeeeee eee
5	3	А потом она сказала aaaaaaaaaaaaaa !!	А потом потом она она сказала aaaaaaaaaaaaaa aaaaaaaaaaaaaa !! !!

Решение

```

var q,i,k,p,f1,f0,j,d:integer;
    s, s1, s2:string;
begin
    readln(s);

    s:=s+' ';
    q:=pos(' ',s);
    s2:="";
    while q>0 do
        begin
            k:=0;
            f0:=1;
            f1:=1;

```

```

s1:=copy(s,1,q-1);
while f1<=length(s1) do
begin
d:=0;
for j:=1 to f1 do begin
if f1 mod j =0 then d:=d+1; end;
if (length(s1)=f1) and (d<=2) then
begin
k:=1;
s2:=s2+s1+' '+s1+' ';
break;
end;
p:=f1;
f1:=f1+f0;
f0:=p;
end;
if k=0 then s2:=s2+s1+' ';
delete(s,1,q);
q:=pos(' ',s);
end;
writeln(s2);
end.

```

Задание 3

В Атлантиде королева по имени Арозора решила запретить все слова, которые одинаково читаются слева направо и справа налево (палиндромы, т.е. слово «атлант» не является палиндромом, а «аталата» – является). Позже, один из советников подсказал ей, что из букв некоторых слов можно составить палиндромы, что привело королеву в ярость. Напишите программу для успокоения королевы, удаляя неудобные ей (с учётом рекомендаций советника) слова из входной строки.

Вводится строка (длиной менее 256 символов), необходимо вернуть строку без запрещённых слов, сохраняя пробелы.

Необходимо написать консольное приложение.

Изменять формат входных и выходных данных запрещено.

Оставлять в конце программы ожидание ввода запрещено.

Помните, что автоматическая проверка населена роботами.

Тесты

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

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

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

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)
```