

Казанский федеральный университет
Очный тур олимпиады для школьников. 2016 год.

Для учащихся 9-х классов

№1. Незнайка учится считать

В Солнечном городе для записи текста используются два знака: * (солнышко) и O (луна – латинская заглавная буква «O»).

Для жителей города Незнайка придумал свою систему знаков для представления цифр, которая формируется из четырех знаков в виде квадрата.

0	1	2	3	4	5	6	7		13	14	15
OO	*O	O*	**	OO	*O	O*	**		*O	O*	**
OO	OO	OO	OO	O*	O*	O*	O*	...	**	**	**

Помогите Незнайке и напишите программу, которая суммирует такие числа.

Входные данные.

В двух строчках записано выражение, где в первой строке два заданных числа разделяются знаком «+», а во второй строке пробелом. Значения не превышают 10^9 .

Выходные данные.

Выходной файл OUTPUT.TXT должен содержать значение суммы, представленной знаками придуманными Незнайкой.

Пример:

INPUT.TXT	OUTPUT.TXT
O**+*OO*	**O
O*OO** **O*	O***O*

Решение.

Считываются 2 строки и по 2 символа разбираются разряды исходного числа в 16-чной системе. Далее, сложив эти числа, производится перевод в 16-чную систему исчисления с записью в 2 строки.

```
Var s1,s2,s,u1,u2:string;
    r,j,c1,c11,c12,c2, c21,c22,c,k:longint;
Begin
Readln(s1);
Readln(s2);
r:=1;
c1:=0; // первое слагаемое
while s1[r]<>'+' do
begin
c11:=ord(s1[r]='*')+2*ord((s1[r+1]='*'));
c12:=8*ord(s2[r]='*')+4*ord((s2[r+1]='*'));
```

```

c1:=c1*16+c11+c12;
r:=r+2
End;
r:=r+1;
c2:=0;
while r<length(s1) do
begin
c21:=ord(s2[r]='*')+2*ord((s1[r+1]='*'));
c22:=8*ord(s2[r]='*')+4*ord((s1[r+1]='*'));
c2:=c2*16+c11+c12;
r:=r+2
End;
C:=c1+c2;
If c=0 then begin writeln('00'); writeln('00') end
else
begin
S1:='';
S2:='';
While c>0 do
begin
r:=c mod 16;
c:=c div 16;
if r mod 2=0 then u1:='O' else u1:='*'; r:=r div 2;
if r mod 2=0 then u1:=u1+'O' else u1:=u1+'*'; r:=r div 2;
if r mod 2=0 then u2:='O' else u2:='*'; r:=r div 2;
if r=0 then u2:=u2+'O' else u2:=u2+'*'; r:=r div 2;
s1:=u1+s1;
s2:=u2+s2
end;
Writeln(s1);
Writeln(s2)
End.

```

№2. Незнайка – художник

Незнайка дружил с Карандашем. И ему нравилось как он рисует.

Незнайка придумал автомат по рисованию на холсте размерности NxM. Этот автомат устанавливался в верхнем левом углу холста с координатой (0,0). Затем Незнайка закладывал капсулу с краской и стрелял из него в координату (X,Y) на

холсте, где отсчет шел от левого верхнего угла. Капсула с краской при полете окрашивала все точки холста, которые оказывались слева и сверху от траектории полета.

Нужно определить размер окрашенной площади после K выстрелов из данного автомата.

Входные данные

В первой строке вводного файла INPUT.TXT заданы размеры холста N, M ($0 < N, M < 1000$) и число выстрелов K ($K < 10^9$).

Далее в K строках заданы координаты выстрелов.

Выходные данные

В выводной файл вывести размер окрашенной площади.

Примеры

№	INPUT.TXT	OUTPUT.TXT
1	1 1 2	2
2	3 1 3 3 1 2 2	6

Решение.

Идея решения .

Заводим матрицу 1000×1000 , заполняя «0».

После каждого выстрела окрашивается площадь прямоугольника $(1, 1, X, Y)$, заполняя все элементы подматрицы «1». Далее надо сосчитать число единиц в этой матрице.

```
Var a:array[0..1000,0..1000] of integer;
```

```
  N,m,i,j,k,x,y:longint;
```

```
Begin
```

```
For i:=1 to 1000 do
```

```
For j:=1 to 1000 do a[i,j]:=0;
```

```
Read(k);
```

```
For m:=1 to k do
```

```
Begin
```

```
Read(x,y);
```

```
For i:=1 to x do
```

```

For j:=1 to y do a[i,j]:=1;
End;
M:=0;
For i:=1 to 1000 do
For j:=1 to 1000 do m:=m+a[i,j];
Write(m)
End.

```

№3. Камыр-батыр

Повар Камыр-батыр в свой день рождения испек вкусный чак-чак и растелил его на круглом столе. Он саблей ловко умел разрубать чак-чак на столе.

Помогите Камыр-батыру посчитать максимальное число гостей, которых он мог бы угостить кусочками чак-чака после N ударов саблей.

INPUT.TXT: *N (0 ≤ N ≤ 1000) – число ударов саблей.*

OUTPUT.TXT: *Число гост ей, кот орых мож ет он угост ит ь.*

Пример:

№	INPUT.TXT	OUTPUT.TXT
1	1	2
2	3	7

Решение.

Идея решения.

После каждого K-го удара появляется K новых кусочков.

```

Var i,n,k:integer;
Begin
Read(n);
K:=1; // в самом начале у нас есть 1 кусок
For i:=1 to n do k:=k+i;
Write(k)
End.

```

№4. Балетмейстер

Балетмейстер Рауф Нури в начале спектакля артистов своей труппы выстраивал на сцене. Для того, чтобы каждый спектакль был уникален, он каждый раз их расставлял по новому.

Рауф Нури придумал следующий алгоритм перебора выхода артистов на сцену.

Всех артистов в начале он построил по росту и, перенумеровав их, вывел на сцену. Далее каждый следующий выход определяется следующим лексикографическим порядком.

Т.е. один порядок расстановки артистов называется большим другого, если при просмотре номеров артистов слева направо первый различающийся номер артиста первом порядке расстановки больше номера соответствующего артиста во втором порядке выхода. (1 2 3 5 4 > 1 2 3 4 5)

Пусть даны два порядка выхода артистов. Вторым порядком выхода артистов является следующим после первого порядка выхода артистов, если второй порядок выхода артистов больше первого и не существует другого порядка выхода артистов, который будет больше первого и меньше второго.

Напишите программу, определяющую по заданному порядку выходов артистов лексикографически следующий порядок выхода.

INPUT.TXT: В первой строке задано N ($0 \leq N \leq 225$) – число артистов. Во второй строке задан текущий порядок выхода артистов.

OUTPUT.TXT: Вывести следующий лексикографический порядок выхода.

Пример:

INPUT.TXT	OUTPUT.TXT
5 1 2 3 5 4	1 2 4 3 5

Решение.

Идея решения:

Программирующие на Си-подобных языках могут использовать функции типа `next_permutation()` для получения следующей перестановки.

Идея написания собственной такой функции состоит в выполнении следующих шагов:

1. Начиная с правого конца, ищем ближайший пик, т.е. число (элемент массива), которое больше обоих своих соседей.

Например, для перестановки 1 3 5 **8** 7 6 4 2 это будет число 8 ($5 < 8 > 7$) в 4-й позиции.

2. Меняем предыдущий элемент для пика (число 5) с ближайшим большим справа от пика (это число 6 в 6-й позиции).

В примере получаем 1 3 **6** 8 7 **5** 4 2

3. Переставляем все числа от пика и правее в порядке возрастания (переворачиваем в обратном порядке) – это следующая перестановка.

В примере получим 1 3 6 2 4 5 7 8

Если первый шаг выполнить нельзя, то это последняя перестановка в лексикографическом порядке. Например, 8 7 6 5 4 3 2 1.

Если последний элемент больше предпоследнего, то меняем их местами и получаем следующую перестановку. Эту операцию нужно выполнять для каждой второй возможной перестановки. Для остальных случаев выполняем описанные выше шаги.

№5. формула

Дана формула, содержащая скобки 3-х видов: круглые, квадратные, фигурные. Формула считается правильной, если в ней правильно расставлены скобки. Другие символы не играют роли.

<строка без скобок> - формула

(формула) – формула

[формула] – формула

{ формула } – формула

формула формула - формула

Проверить правильность расстановки скобок и напечатать ПРАВИЛЬНО или НЕ ПРАВИЛЬНО.

Входные данные

Дана - символьная строка, содержащая некоторую формулу.

Выходные данные

Напечатать ПРАВИЛЬНО или НЕ ПРАВИЛЬНО в зависимости от правильности формулы.

Пример:

№	INPUT.TXT	OUTPUT.TXT
1	(a+b)*[{}()]	ПРАВИЛЬНО
2	3+((a-c)]	НЕПРАВИЛЬНО

Решение.

Идея решения:

Используем стек (stack) для хранения открывающих скобок. Изначально он пустой.

Когда появляется открывающая скобка, она автоматически попадает в вершину стека.

Когда приходит какая-то закрывающая скобка, то в правильной формуле на вершине стека должна храниться соответствующая ей открывающая скобка того же вида. Если в этот момент стек пуст или скобка в вершине другого типа, то это свидетельствует об ошибке в формуле – закрывающая скобка не соответствует открывающей. Если скобка соответствующая, то она из стека удаляется.

Кроме того, в конце формулы стек должен опустеть, т.е. все открытые скобки должны быть закрыты.

№6. Номенклатура

В некоторой компании работают N сотрудников, пронумерованных от 1 до N . У каждого сотрудника есть ровно 1 непосредственный начальник, которому он подчинён.

Сотрудник также подчинён все другим начальникам, которым подчинён его непосредственный начальник.

Только у Самого Главного Босса нет начальников, есть только подчинённые.

Вычислить количество подчиненных у каждого сотрудника

Входные данные

В первой строке дано количество сотрудников N ($N < 1000$) и во второй строке $N-1$ пара номеров сотрудников A и B (сотрудник A подчинён сотруднику B , т.е. B - непосредственный начальник для A).

Выходные данные

Напечатать количество подчиненных у каждого из сотрудников по порядку.

Пример:

INPUT.TXT	OUTPUT.TXT
6 2 1 3 1 4 3 5 3 6 5	5 0 3 0 1 0

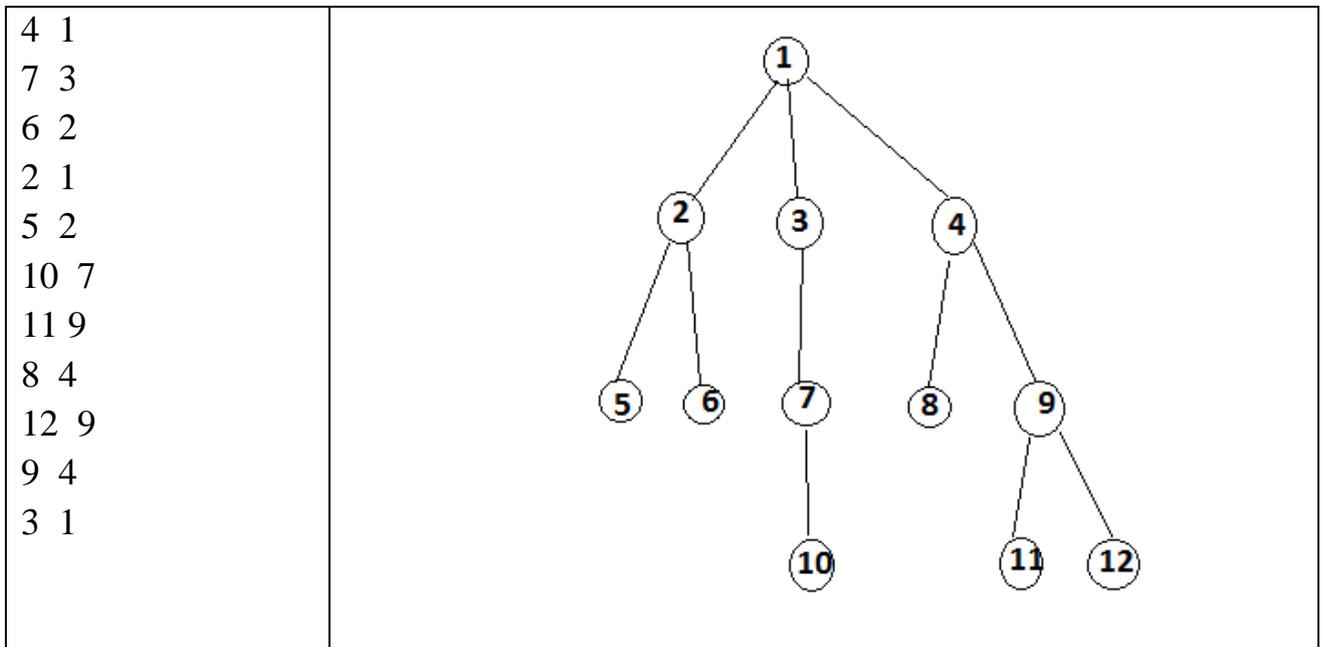
Решение.

Идея решения:

Очевидно, что все сотрудники образуют дерево с корнем - Самый Главный Босс. На каждом уровне размещаются подчинённые сотрудников предыдущего уровня. На рисунке 1 приведён пример дерева для фирмы из 12 сотрудников (номера от 1 до 12) с приведённым ниже списком подчинённости. Пары Подчинённый – Начальник приведены в произвольном порядке.

$N = 12$

Данные	дерево
--------	--------



После этого, начиная с Самого Главного Босса (у него нет начальников!), последовательно по уровням помещаем каждого сотрудника в очередь. Порядок помещения в очередь сотрудников одного уровня не важен.

Очередь: 1 (0) 2 (0) 3 (0) 4 (0) 5 (0) 6 (0) 7 (0) 8 (0) 9 (0) 10 (0) 11 (0) 12 (0)

Далее будем просматривать построенную очередь в обратном порядке и приписывать каждому сотруднику число – количество его подчинённых. Изначально все получают 0.

Число у каждого сотрудника +1 прибавляем его непосредственному начальнику.

12 (0) 11 (0) 10 (0) 9 (2) 8 (0) 7 (1) 6 (0) 5 (0) 4 (4) 3 (2) 2 (2) 1 (11)

Для быстрой реализации будем хранить всех подчинённых каждого сотрудника в виде линейного списка. Кроме того, в отдельном массиве будем хранить начальника для каждого сотрудника.

Казанский федеральный университет
Очный тур олимпиады для школьников. 2016 год.

Для учащихся 10-х классов

№1. Незнайка учится считать

В Солнечном городе для записи текста используются четыре знака: * (солнышко), O (луна – латинская заглавная буква «O») и «/», «\» (лучи).

Для жителей города Незнайка придумал свою систему знаков для представления цифр, которая формируется из четырех знаков в виде квадрата.

	Левая сторона	Правая сторона	Примеры записи чисел	Запись числа Незнайкой	В 10-чной системе		
Верхний уровень	O - 0 / - 1 * - 2	O - 0 \ - 3 * - 6	1	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>**</td></tr><tr><td>\ /</td></tr></table>	**	\ /	44
**							
\ /							
Нижний уровень	O - 0 \ - 27 * - 54	O - 0 / - 9 * - 18	2	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>/*</td></tr><tr><td>*</td></tr></table>	/*	*	52
/*							
*							

Помогите Незнайке и напишите программу, которая суммирует такие числа.

Входные данные.

В двух строчка записано выражение, где в первой строке два заданных числа разделяются знаком «+», а во второй строке пробелом. Значения не превышают 10^9 .

Выходные данные.

Выходной файл OUTPUT.TXT должен содержать значение суммы, представленной знаками придуманными Незнайкой.

Пример:

INPUT.TXT	OUTPUT.TXT
/**\+O*O	/O/*
**/ O*\O	**O/

Решение.

Считываются 2 строки и по 2 символа разбираются разряды исходного числа в 81-чной системе. Далее, сложив эти числа, производится перевод в 16-чную систему исчисления с записью в 2 строки.

```

Var s1,s2,s,u1,u2:string;
    r,j,c1,c11,c12,c2, c21,c22,c,k:longint;
Begin
Readln(s1);
Readln(s2);
r:=1;
c1:=0; // первое слагаемое
while s1[r]<>'+' do

```

```

begin
c11:=ord(s1[r]='*')+2*ord((s1[r+1]='*'));
c12:=8*ord(s2[r]='*')+4*ord((s2[r+1]='*'));
c1:=c1*81+c11+c12;
r:=r+2
End;
r:=r+1;
c2:=0;
while r<length(s1) do
begin
c21:=ord(s2[r]='*')+2*ord((s1[r+1]='*'));
c22:=8*ord(s2[r]='*')+4*ord((s1[r+1]='*'));
c2:=c2*81+c11+c12;
r:=r+2
End;
C:=c1+c2;
If c=0 then begin writeln('00'); writeln('00') end
else
begin
S1:='';
S2:='';
While c>0 do
begin
r:=c mod 81;
c:=c div 81;
if r mod 3=0 then u1:='O' else u1:='*'; r:=r div 2;
if r mod 3=0 then u1:=u1+'O' else u1:=u1+'*'; r:=r div 2;
if r mod 3=0 then u2:='O' else u2:='*'; r:=r div 2;
if r=0 then u2:=u2+'O' else u2:=u2+'*'; r:=r div 2;
s1:=u1+s1;
s2:=u2+s2
end;
Writeln(s1);
Writeln(s2)
End.

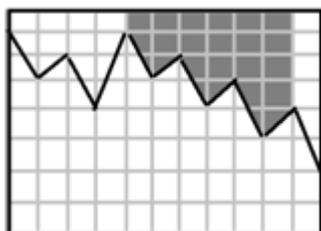
```

№2. Самая длинная К-пила

Незнайка любил наблюдать как мастерит Самоделкин. Он тоже решил для общего дела сделать из листа стали размера $N \times M$ К-пилу, у которой высота зубьев не превышает K .

Он взял у Самоделкина сталерезательный станок, который прорезает стальные листы последовательно слева направо от одной заданной целой координаты до другой. Координаты задаются последовательно высотами точек

Нужно определить, какой длины окажется самая длинная и легкая К-пила, полотно которой вырезается по вертикали с целыми координатами. Легкая К-пила выбирается среди самых длинных и минимальной площадью полотна.



Входные данные

В первой строке задается размер полотна $N \times M$ ($0 < N < 1000$, $1 < M < 100$) и допустимая высота зуба пилы K ($0 < K < M$).

Далее N высот точек, через которые Незнайка пилит полотно.

Входные данные

Выводится два числа: максимальная длина пилы и минимальная площадь полотна пилы.

Пример

№	INPUT.TXT	OUTPUT.TXT
1	7 11 2 1 3 2 4 1 3 2 4 3 5 4 6	3 19.5
2	3 6 2 2 1 3 1 2 1 2	6 5.5

Решение.

Идея решения.

Рассматриваются две последовательности высот координат зубьев;

$X_1, X_2, X_3, X_4, X_5, X_6, \dots$

$M - X_1, M - X_2, M - X_3, M - X_4, M - X_5, M - X_6, \dots$

Для этих двух последовательностей находится максимальная к-пила и выбирается максимум из этих двух пил.

Алгоритм поиска К-пилы для последовательности имеет следующий вид:

```
var a,b,z,r,i,n,k:longint;  
    m:boolean;  
begin  
    read(n,k);  
    case n of  
        0..2:write(n)  
    else  
        read(a,b);  
        m:=a<b;r:=2;z:=2;  
        for i:=3 to n do  
            begin  
                a:=b;  
                read(b);  
                if m and (a>b) and (a-b>=k) or not m and (a<b) and (a-b<=k) then  
                    begin  
                        r:=r+1;  
                        m:=not m  
                    end  
                else  
                    r:=2;  
                if r>z then z:=r;  
            end;  
            write(z)  
        end  
    end.
```

№3. Камыр-батыр

Повар Камыр-батыр в свой день рождения испек N вкусных перемячей. И всех желающих угасал этими перемячами. Для того чтобы перемячи не остыли, он первым желающим давал по многу штук, а тем, кто просил позже, давал меньше. Если бы у него было 6 перемячей, то он мог бы раздать 4-мя способами: $3+2+1$ (первого любителя 3 перемячами, второго 2-мя, а третьего 1-им перемячем угостил бы), $4+2$, $5+1$, 6.

Напишите программу, которая помогла бы посчитать Камыр-батыру число возможных раздач перемячей, если бы он испек N вкусных перемячей.

INPUT.TXT: N ($0 \leq N \leq 225$) – число перемячей.

OUTPUT.TXT: Число способов раздачи перемячей.

Пример:

INPUT.TXT	OUTPUT.TXT
6	4

Решение.

```
def f(n) :  
    if n < 3 : return 1  
    s = 0  
    for i in range(1, ((n-1)//2+1) :  
        s += f(i)  
    return s
```

№4. Балетмейстер

Балетмейстер Рауф Нури в начале спектакля артистов своей труппы выстраивал на сцене. Для того чтобы каждый спектакль был уникален, он их расставлял каждый раз по новому.

Рауф Нури придумал следующий алгоритм перебора выхода артистов на сцену.

Всех артистов вначале он построил по росту и, перенумеровав, вывел на сцену. Далее каждый следующий выход определяется следующим лексикографическим порядком.

Т.е. один порядок расстановки артистов называется большим другого, если при просмотре номеров артистов слева направо первый различающийся номер артиста в первом порядке расстановки больше номера соответствующего артиста во втором порядке выхода. (1 2 3 5 4 > 1 2 3 4 5)

Пусть даны два порядка выхода артистов. Второй порядок выхода артистов является следующим после первого порядка выхода артистов, если второй порядок выхода артистов больше первого и не существует другого порядка выхода артистов, который будет больше первого и меньше второго.

Напишите программу нахождения порядка артистов на K -й день

INPUT.TXT: В первой строке задано N ($0 \leq N \leq 225$) – число артистов и K -й день ($0 \leq K \leq 365$).

OUTPUT.TXT: Вывест и K -й лексикографический порядок выхода.

Пример:

INPUT.TXT	OUTPUT.TXT
5 14	1 4 2 5 3

Решение.

Идея решения:

Программирующие на Си-подобных языках могут использовать функции типа `next_permutation ()` для получения следующей перестановки.

Идея написания собственной такой функции состоит в выполнении следующих шагов:

1. Начиная с правого конца, ищем ближайший пик, т.е. число (элемент массива), которое больше обоих своих соседей.

Например, для перестановки `1 3 5 8 7 6 4 2` это будет число `8 (5 < 8 > 7)` в 4-й позиции.

2. Поменять предыдущий элемент для пика (число `5`) с ближайшим элементом большим справа от пика (это число `6` в 6-й позиции).

В примере получаем `1 3 6 8 7 5 4 2`

3. Переставляем все числа от пика и правее в порядке возрастания (переворачиваем в обратном порядке) – это следующая перестановка.

В примере получим `1 3 6 2 4 5 7 8`

Если первый шаг выполнить нельзя, то это последняя перестановка в лексикографическом порядке. Например, `8 7 6 5 4 3 2 1`.

Если последний элемент больше предпоследнего, то меняем их местами и получаем следующую перестановку. Эту операцию нужно выполнять для каждой второй возможной перестановки. Для остальных случаев выполняем описанные выше шаги.

№5. Формула

Дана формула, содержащая скобки 3-х видов: круглые, квадратные, фигурные. Формула считается правильной, если в ней правильно расставлены скобки, а другие символы не играют роли.

Проверить правильность расстановки скобок и напечатать ПРАВИЛЬНО или НЕ ПРАВИЛЬНО.

Правила:

<строка без скобок>

(Ф)

[Ф]

{Ф}

ФФ

Входные данные

Дана - символьная строка, содержащая некоторую формулу.

Выходные данные

Напечатать два минимальных числа левых и правых скобок, необходимых для исправления.

Пример:

№	INPUT.TXT	OUTPUT.TXT
1	(a+b)*[()]	ПРАВИЛЬНО
2	{(3)+(d)}((a+f)+1)(b+[a-c])	НЕ ПРАВИЛЬНО

Решение.

В стек (stack) записываются открывающие скобки. Изначально он пустой.

Когда появляется открывающая скобка, она автоматически попадает в вершину стека.

Когда приходит какая-то закрывающая скобка, то в правильной формуле на вершине стека должна храниться соответствующая ей открывающая скобка того же вида. Если в этот момент стек пуст или скобка в вершине другого типа, то это свидетельствует об ошибке в формуле – закрывающая скобка не соответствует открывающей. Если скобка соответствующая, то она из стека удаляется.

Кроме того, в конце формулы стек должен опустеть, т.е. все открытые скобки должны быть закрыты.

№6. Китайская стена

Даны кубики n цветов и количество кубиков заданного цвета a_1, a_2, \dots, a_n , нужно составить башню максимальной высоты так, чтобы выполнялись два условия:

- кубики i -го цвета встречались в башне не больше a_i раз;
- количество вхождений в башню кубиков каждого из цветов должно быть **различным** для всех цветов, которые встретились в башне хотя бы один раз.

Входные данные

В первой строке входных данных записано единственное целое число n ($2 \leq n \leq 100$) — количество цветов.

В следующей строке записано n целых чисел a_i ($1 \leq a_i \leq 10^9$) — ограничение на количество вхождений кубиков i -го цвета в башню.

Выходные данные

Выведите одно целое число — максимальную высоту башни, удовлетворяющей требованиям.

Примеры:

№	INPUT.TXT	OUTPUT.TXT
1	3 2 5 5	11
2	3 1 1 2	3

Примечание

Для удобства возьмём три цвета: «r», «g», «b» (красный, зеленый синий). В первом тесте из условия, примерами подходящих башен являются: «bbbrrrrgbbgg», «rrgbgbgbgbg». Во втором тесте из условия примерами подходящих строк являются: «rbb», «bgb».

Решение.

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.PrintWriter;
import java.util.Arrays;
import java.util.StringTokenizer;
public class chinese_wall {
    public static void main(String[] args) throws IOException {
        BufferedReader in = new BufferedReader(new FileReader("INPUT.TXT"));
        init(in);
        in.close();
        solve();
        PrintWriter out = new PrintWriter("OUTPUT.TXT");
        print(out);
        out.close();
    }
    public static int n;
    public static int[] w;
    public static long ans;
    private static void init(BufferedReader in) throws IOException {
        n = Integer.parseInt(in.readLine().trim());
        StringTokenizer st = new StringTokenizer(in.readLine().trim());
        w = new int[n];
        for (int i = 0; i < n; i++){
            w[i] = Integer.parseInt(st.nextToken());
        }
    }
    private static void solve() {
        ans = 0;
        Arrays.sort(w);
        for (int i = n-1; i >= 0; i--){
            if (w[i] == 0) break;
            ans += w[i];
            for (int j = i-1; j >= 0; j--){
```

```
        if (w[j] == w[i]) w[j]--;  
    }  
}  
}  
private static void print(PrintWriter out) throws IOException {  
    out.print(ans);  
}  
}
```

Казанский федеральный университет
Очный тур олимпиады для школьников. 2016 год.

Для учащихся 11-х классов

Ограничения по времени 1 с.
 Ограничения по памяти 256 МБ.

№1. Незнайка учится считать

В Солнечном городе для записи текста используются четыре знака: * (солнышко) и O (луна –латинская заглавная буква), «/», «\» (лучи). Жители этого города умеют только складывать и умножать.

Для жителей города Незнайка придумал свою систему знаков для представления цифр, которая формируется из четырех знаков в виде квадрата.

Квадрат Незнайки	Левая сторона	Правая сторона	Примеры записи цифр	Запись цифр Незнайкой	В 10-чной системе
Верхний уровень	O - 0 / - 1 * - 2	O - 0 \ - 3 * - 6	1		44
Нижний уровень	O - 0 \ - 27 * - 54	O - 0 / - 9 * - 18	2		52

Помогите Незнайке и напишите программу, которая суммирует и умножает такие числа.

Входные данные.

В двух строчка записано выражение, где в первой строке два числа разделяются знаком «+» или «X », а во второй строке пробелом. Значения чисел не превышают 10^{18} .

Выходные данные.

Выходной файл OUTPUT.TXT должен содержать значение суммы, представленной знаками придуманными Незнайкой.

Пример:

№	INPUT.TXT	OUTPUT.TXT
1	/**\+O*O **/ O*\O	/O/* **O/
2	O\O\O\O\ O/O O/	/O/O\ OO*OO/

Решение.

Считываются 2 строки и по 2 символа разбираются разряды исходного числа в 81-чной системе. Далее, сложив эти числа, производится перевод в 16-чную систему исчисления с записью в 2 строки.

```
Var s1,s2,s,u1,u2:string;
    r,j,c1,c11,c12,c2, c21,c22,c,k:longint;
    op:char;
```

```

Begin
Readln(s1);
Readln(s2);
r:=1;
c1:=0; // первое слагаемое
while( s1[r]<>'+' ) and (s1[r]<>'X') do
begin
c11:=ord(s1[r]='*')+2*ord((s1[r+1]='*'));
c12:=8*ord(s2[r]='*')+4*ord((s2[r+1]='*'));
c1:=c1*81+c11+c12;
r:=r+2
End;
Op:= s1[r];
r:=r+1;
c2:=0;
while r<length(s1) do
begin
c21:=ord(s2[r]='*')+2*ord((s1[r+1]='*'));
c22:=8*ord(s2[r]='*')+4*ord((s1[r+1]='*'));
c2:=c2*81+c11+c12;
r:=r+2
End;
If op='+' then C:=c1+c2
Else; C:=c1*c2
If c=0 then begin writeln('00'); writeln('00') end
else
begin
S1:='';
S2:='';
While r>0 and c>0 do
begin
r:=c mod 81;
e:=c mod 81; c:=c div 81 ;
if r mod 3=0 then u1:='O' else u1:='*'; r:=r div 2;
if r mod 3=0 then u1:=u1+'O' else u1:=u1+'*'; r:=r div 2;
if r mod 3=0 then u2:='O' else u2:='*'; r:=r div 2;
if r=0 then u2:=u2+'O' else u2:=u2+'*'; r:=r div 2;
s1:=u1+s1;
s2:=u2+s2

```

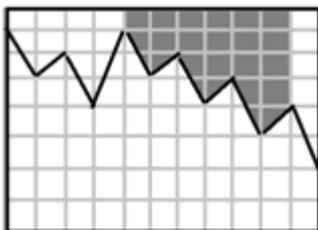
```
end;
Writeln(s1);
Writeln(s2)
End.
```

№2. Самая длинная К-пила

Незнайка любил наблюдать как мастерит Самоделкин. Он тоже решил для общего дела сделать из листа стали размера $N \times M$ К-пилу, у которой высота зубьев не превышает K .

Он взял у Самоделкина сталерезательный станок, который прорезал стальные листы последовательно слева направо от одной целой координаты до другой. Координаты задаются последовательно высотами точек

Нужно определить, какой длины окажется самая длинная и легкая К-пила, полотно которой вырезается по вертикали с целыми координатами. Легкая К-пила выбирается среди самых длинных и минимальной площадью полотна.



Входные данные

В первой строке задается размер полотна $N \times M$ ($0 < N < 10^9$, $1 < M < 1000$) и допустимая высота зуба пилы K ($0 < K < M$).

Далее N высот точек, через которые Незнайка пилит полотно.

Входные данные

Выводится два числа: максимальная длина пилы и минимальная площадь полотна пилы.

Пример

№	INPUT.TXT	OUTPUT.TXT
1	11 7 2 1 3 2 4 1 3 2 4 3 5 4 6	3 19.5
2	6 3 2 2 1 3 1 2 1 2	6 5.5

Решение.

Идея решения.

Рассматриваются две последовательности высот координат зубьев;

$X_1, X_2, X_3, X_4, X_5, X_6, \dots$

$M-X_1, M-X_2, M-X_3, M-X_4, M-X_5, M-X_6, \dots$

Для этих двух последовательностей находится максимальная k -пила и выбирается максимум из этих двух пил.

Алгоритм поиска K -пилы для последовательности имеет следующий вид:

```
var a,b,z,r,i,n,k:longint;  
    m:boolean;  
begin  
    read(n,k);  
    case n of  
        0..2:write(n)  
    else  
        read(a,b);  
        m:=a<b;r:=2;z:=2;  
        for i:=3 to n do  
            begin  
                a:=b;  
                read(b);  
                if m and (a>b) and (a-b>=k) or not m and (a<b) and (a-b<=k) then  
                    begin  
                        r:=r+1;  
                        m:=not m  
                    end  
                else  
                    r:=2;  
                if r>z then z:=r;  
            end;  
        write(z)  
    end  
end.
```

№3. Камыр-батыр

Повар Камыр-батыр в свой день рождения решил угостить своих гостей разными вкусными многослойными пирогами. Он пошел на базар и закупил N различных приправ.

Напишите программу, которая помогла бы посчитать Камыр-батыру число пирогов, которые он мог бы приготовить из этих приправ.

INPUT.TXT: N ($0 \leq N \leq 225$) – число приправ.

OUTPUT.TXT: Число различных пирогов.

Пример:

INPUT.TXT	OUTPUT.TXT
3	4
4	11

Решение.

```
n=int(input()) #PYTHON
```

```
print(2**n-1-n)
```

№4. Балетмейстер

Балетмейстер Рауф Нури в начале спектакля артистов своей труппы выстраивал на сцене. Для того чтобы каждый спектакль был уникален, он их расставлял каждый раз по новому.

Рауф Нури придумал следующий алгоритм перебора выхода артистов на сцену.

Всех артистов в начале он построил по росту и, перенумеровав, вывел на сцену. Далее каждый следующий выход определяется следующим лексикографическим порядком.

Т.е. один порядок артистов называется большим другого, если при просмотре номеров артистов слева направо первый различающийся номер артиста первого порядка больше номера соответствующего артиста второго порядка. (1 2 3 5 4 > 1 2 3 4 5)

Если даны два порядка выхода артистов. Второй порядок выхода артистов является следующим после первого порядка выхода артистов, если второй порядок выхода артистов больше первого и не существует другого порядка выхода артистов, который будет больше первого и меньше второго.

Напишите программу нахождения порядка артистов на K -й день

INPUT.TXT: В первой строке задано N ($0 \leq N \leq 1000$) – число артистов и K -й день ($0 \leq K \leq 1000$).

OUTPUT.TXT: Вывест и K -й лексикографический порядок выхода.

Пример:

INPUT.TXT	OUTPUT.TXT
5 14	1 4 2 5 3

Решение.

Идея решения:

Программирующие на Си-подобных языках могут использовать функции типа `next_permutation()` для получения следующей перестановки.

Идея написания собственной такой функции состоит в выполнении следующих шагов:

1. Начиная с конца, ищем ближайший пик, т.е. число (элемент массива), которое больше обоих своих соседей.

Например, для перестановки 1 3 5 **8** 7 6 4 2 это будет число 8 ($5 < 8 > 7$) в 4-й позиции.

2. Поменять предыдущий элемент для пика (число 5) с ближайшим большим справа от пика (это число 6 в 6-й позиции).

В примере получаем 1 3 **6** 8 7 **5** 4 2

3. Переставляем все числа от пика и правее в порядке возрастания (переворачиваем в обратном порядке) – это следующая перестановка.

В примере получим 1 3 6 **2 4 5 7 8**

Если первый шаг выполнить нельзя, то это последняя перестановка в лексикографическом порядке. Например, 8 7 6 5 4 3 2 1.

Если последний элемент больше предпоследнего, то меняем их местами и получаем следующую перестановку. Эту операцию нужно выполнять для каждой второй возможной перестановки. Для остальных случаев выполняем описанные выше шаги.

№5. Формула

Дана формула, содержащая круглые скобки.

Формула считается правильной, если в ней правильно расставлены скобки, а другие символы не играют роли. Если формула не правильная, то нужно дописать слева и справа необходимое минимальное число открывающих и закрывающих скобок

<строка без скобок> - формула
(формула) – формула
формула формула - формула

Проверить правильность расстановки скобок и напечатать два числа скобок открывающих и закрывающих, необходимых для исправления ошибок.

Входные данные

Дана - символьная строка, содержащая некоторую формулу.

Выходные данные

Напечатать два минимальных числа левых и правых скобок, необходимых для исправления.

Пример:

№	INPUT.TXT	OUTPUT.TXT
1	(a+b)*()	0 1
2	3)+(d)((a+f)+1))(b+(a-c)	3 2

Решение.

Идея решения:

Используем число открытых левых скобок.

Когда появляется открывающая скобка, она автоматически увеличивает счетчик. При появлении правой скобки счетчик скобок уменьшается. Если число открытых левых скобок становится -1, то он обнуляется, а число дополнительных левых скобок увеличивается на 1. В конце число не парных оставшихся левых скобок определяют необходимое число правых скобок.

```
Var r,l,dr,dl,i,j,k:integer; c:char;
```

```
Begin
```

```
R:=0;l:=0;dr:=0;dl:=0;
```

```
Repeat
```

```
Read( c);
```

```
Case c of
```

```
‘(’:l:=l+1;
```

```
‘)’: if l=0 then dl:=dl+1 else l:=l-1
```

```
End;
```

```
Until eof;
```

```
Write(dl,’ ‘,dr)
```

```
End.
```

№6. Маугли – строитель

Джунгли — это место, где есть период, когда постоянно идут проливные дожди. Маугли решил помочь своим друзьям и построить навесы для зверей. Для этого он выбрал часть джунглей, в которой на одной прямой растут N деревьев. У i -го дерева высота целое положительное число h_i , а располагается в точке с координатой x_i . Маугли строит навес следующим образом: он выбирает максимально число деревьев одной высоты, так чтобы между ними не было деревьев выше, и строит крышу длиной от самого левого дерева из этого набора до самого правого. Потом повторяет ту же процедуру еще раз, пока ни останутся только одиноко торчащие деревья без крыши. Отметим, что крыш может быть несколько на нескольких уровнях, специально для защиты особо важных участков джунглей.

Помогите Маугли определить какая будет суммарная длина всех крыш, чтобы он мог рассчитать сколько материалов ему придется использовать.

Входные данные

В первой строке вводного файла INPUT.TXT задано количество деревьев N ($0 < N < 100000$). В следующих N строках записаны пары целых чисел x_i и h_i разделенные пробелом, которые описывают i -ое дерево ($0 < x_i$ и $h_i < 10^9$).

Выходные данные

В выводной файл вывести одно единственное число — суммарную длину всех крыш.

Примеры

№	INPUT.TXT	OUTPUT.TXT	Описание
1	5 1 6	7	Есть три дерева высоты 6 с номерами 1,2,4, между ними нет деревьев выше них. Самое левое дерево с координатой 1,

	5 6 2 3 7 6 3 3		самое правое с координатой 7, над ними построят крышу длинны 6. И еще два дерева высоты 3 с номерами 3 и 5, между ними тоже нет деревьев выше них, у них координаты 2 и 3, а значит над ними строится крыша длинны 1. Итого суммарная длинна крыш равна 7
2	6 1 100 10 1 12 100 7 1 20 100 15 1	22	Есть три дерева высоты 100 с номерами 1,3,5, между ними нет деревьев выше них. Самое левое дерево с координатой 1, самое правое с координатой 20, над ними построят крышу длинны 19. И еще два дерева высоты 1 с номерами 2 и 4, дерево с номером 6 к ним присоединить нельзя, так как между 6-м и 2-м есть 3-е дерево, которое мешает. В результате мы строим крышу между 2-м и 4-м деревьями длинной 3, и 6-ое дерево останется одиноким без крыши. Итого суммарная длинна 22.

Решение.

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.PrintWriter;
import java.util.Arrays;
import java.util.StringTokenizer;
public class kfuabitur_maugli {
    public static void main(String[] args) throws IOException {
        BufferedReader in = new BufferedReader(new FileReader("INPUT.TXT"));
        init(in);
        in.close();
        solve();
        PrintWriter out = new PrintWriter("OUTPUT.TXT");
        print(out);
        out.close();
    }
    public static int n;
    public static Tree[] a;
    public static long ans;
    public static int INF = Integer.MAX_VALUE;
    private static void init(BufferedReader in) throws IOException {
        n = Integer.parseInt(in.readLine().trim());
        a = new Tree[n+1];
        for (int i = 0; i < n; i++){
            StringTokenizer st = new StringTokenizer(in.readLine().trim());
            a[i] = new Tree(Integer.parseInt(st.nextToken()),
                Integer.parseInt(st.nextToken()));
        }
    }
    private static void solve() {
        // ...
    }
    private static void print(PrintWriter out) {
        out.println(ans);
    }
}
```

```

    }
}
private static void solve() {
    a[n] = new Tree(INF, INF);
    Arrays.sort(a);
    Tree[] stack = new Tree[n];
    int stackSize = 1;
    stack[0] = new Tree(-10, INF);
    for (int i = 0; i <n+1; i++){
        if (stack[stackSize-1].h < a[i].h){
            int h = stack[stackSize-1].h;
            int xb = stack[stackSize-1].x;
            int xe = xb;
            stackSize--;
            while (stackSize>1 && stack[stackSize-1].h<a[i].h){
                if (stack[stackSize-1].h!=h){
                    ans+=(xb-xe);
                    h = stack[stackSize-1].h;
                    xb = stack[stackSize-1].x;
                    xe = xb;
                }
                else{ xe = stack[stackSize-1].x; }
                stackSize--;
            }
            ans+=(xb-xe);
        }
        stack[stackSize] = a[i];
        stackSize++;
    }
}
private static void print(PrintWriter out) throws IOException {
    out.print(ans);
}
static class Tree implements Comparable<Tree>{
    int x,h;
    public Tree(int x, int h) {
        this.x = x;
        this.h = h;
    }
}

```

```
@Override
public int compareTo(Tree o) {
    return Integer.compare(this.x, o.x);
}
}
```