**Задачи олимпиады по информатике для 6 - 8 классов.**

Ввод исходных данных с клавиатуры, вывод на экран, ограничение памяти 256 Мб, времени 1 с.

**Задача 1. Шахматная доска (100 баллов)**

Шахматная доска состоит из *n* × *m* клеток, покрашенных в черный и белый цвет в «шахматном» порядке. При этом клетка в левом нижнем углу доски покрашена в черный цвет. Определите, сколько всего на доске черных клеток.

**Входные данные**

Вводятся два числа *n* и *m*, записанных в одной строке. Все числа — натуральные, не превосходящие 32 000.

**Выходные данные**

Вывести одно целое число — количество черных клеток на доске.

|  |  |  |  |
| --- | --- | --- | --- |
| **Входные данные** | Пример 1 | Пример 2 | Пример 3 |
| N М | 3 4 | 8 8 | 5 7 |
| **Выходные данные** |  |  |  |
| К | 6 | 32 | 18 |

**Задача 2. Строки в книге (100 баллов)**

В книге на одной странице помещается K строк. Таким образом, на 1-й странице печатаются строки с 1-й по K-ю, на второй — с (K+1)-й по (2∙K)-ю и т.д. Напишите программу, которая по номеру строки в тексте определяет номер страницы, на которой будет напечатана эта строка, и порядковый номер этой строки на странице.

**Входные данные**

Вводятся два числа: K — количество строк, которое печатается на странице, и N — номер строки (1≤K≤200, 1≤N≤20000).

**Выходные данные**

Выведите два числа — номер страницы, на которой будет напечатана эта строка, и номер строки на странице.

|  |  |  |  |
| --- | --- | --- | --- |
| **Входные данные** | Пример 1 | Пример 2 | Пример 3 |
| К N  | 50 1 | 20 25 | 15 43 |
| **Выходные данные** |  |  |  |
| C M | 1 1  | 2 5 | 3 13 |

**Задача 3. Грибники (100 баллов)**

N (2 <= N <= 20 000) грибников пошли в лес за грибами. Каждый нашел b[i] грибов (целые числа в диапазоне от 1 до 30 000). Найдите номер грибника, который нашел больше всех грибов и какое минимальное количество грибов нужно насобирать грибнику, который насобирал меньше всех, чтобы обогнать самого лучшего грибника? Считаем, что только один грибник насобирал меньше всех грибов.

# **Входные данные:**

# Первая строка ввода содержит число N.

# Вторая строка ввода содержит N чисел b[1], b[2], ... b[N] , которые вводятся через пробел.

# **Выходные данные:**

К - номер грибника, который нашел больше всех грибов.

М - минимальное количество грибов, которое нужно насобирать грибнику, который насобирал меньше всех, чтобы обогнать самого лучшего грибника.

|  |  |  |  |
| --- | --- | --- | --- |
| **Входные данные** | Пример 1 | Пример 2 | Пример 3 |
| N  b[1] b[2] ... b[N] | 4 5 2 3 1 | 63 8 2 5 4 5 | 84 5 6 5 4 2 1 9 |
| **Выходные данные** |  |  |  |
| К М | 1 5 | 2 7 | 8 9 |

**Задача 4. Накопление строки (100 баллов)**

Вводится символ С и строка S.

Исходное состояние новой строки P - символ C.

Далее новая строка P достраивается по правилу: Если очередной символ строки S меньше первого символа (С) строки P, то он добавляется в начало строки P иначе – в конец строки P.

# **Входные данные:**

Первая строка ввода содержит один символ С.

Вторая строка содержит строковую величину S с длиной ≤ 255.

# **Выходные данные:**

Вывод должен содержать одну строковую величину P.

|  |  |  |  |
| --- | --- | --- | --- |
| **Входные данные** | Пример 1 | Пример 2 | Пример 3 |
| СS | CACTG | ACT | TCХTA |
| **Выходные данные** |  |  |  |
| P | ACCTG | ACT | ACTХT |

Пояснение: в строковых величинах буквы можно сравнивать между собой, как и цифры. Большей считается буква, стоящая в алфавите дальше, т.е. буква В >(больше) буквы А.

**Задача 5. Числа (100 баллов)**

Саша и Катя учатся в начальной школе. Для изучения арифметики при этом используются карточки, на которых написаны цифры (на каждой карточке написана ровно одна цифра). Однажды они пришли на урок математики, и Саша, используя все свои карточки, показал число *A*, а Катя показала число *B*. Учитель тогда захотел дать им такую задачу, чтобы ответ на нее смогли показать и Саша, и Катя, каждый используя только свои карточки. При этом учитель хочет, чтобы искомое число было максимально возможным.

**Входные данные**

Вводятся два целых неотрицательных числа *A* и *B* (каждое число в одной строке). Длина каждого из чисел не превосходит 200 цифр.

**Выходные данные**

Выведите одно число — максимальное целое число, которое можно составить, используя как цифры первого числа, так и цифры второго числа. Если же ни одного такого числа составить нельзя, выведите -1.

|  |  |  |  |
| --- | --- | --- | --- |
| **Входные данные** | Пример 1 | Пример 2 | Пример 3 |
| АВ | 280138798081 | 12389456 | 268913698 |
| **Выходные данные** |  |  |  |
| С | 8810 | -1 | 986 |

**Задача 1.**

Если на доске — четное число клеток, то черных и белых клеток поровну, поэтому

ответом будет *nm*/2. Если же произведение *nm* нечетно, то черных клеток будет на одну больше, поэтому можно поделить *nm* на 2 нацело и прибавить к результату 1.

**Program** z1;

**var**

n, m : integer;

 k, s: longint;

**begin**

read(n,m);

 s:=n\*m;

 **if** s **mod** 2 = 0

 **then** k := s **div** 2

 **else** k := (s **div** 2)+1;

 write(k);

**end**.

**Задача 2.**

**rogram** z2;

**var**

k,n,m,c : integer;

 **begin**

read(k,n);

 c:= (n **div** k)+1;

 m:= n **mod** k;

 **if** m=0 **then begin** m:=k;c:=c-1;**end**;

 write(c,' ',m);

**end**.

**Задача 3.**

**program** z3;

 **var** b: **array**[1..20000] **of** integer;

 i, n, k, m, max, min : integer;

**begin**

readln(n);

 **for** i := 1 **to** n **do** read(b[i]);

 max:=b[1]; min:=b[1]; k:=1;

 **for** i := 2 **to** n **do**

 **begin**

 **if** b[i]>max

 **then begin** k:=i;

 max:=b[i];

 **end**;

 **if** b[i]<min

 **then** min:=b[i];

 **end**;

 m:=max-min+1;

 writeln (k,' ',m);

**End**.

**Задача 4.**

**program** z4;

 **var** s,p :string; c:char;

 i, n: integer;

**begin**

readln(c);

 readln(s);

 n:=length(s);

 p:=c;

 **for** i := 1 **to** n **do**

 **if** s[i]<c

 **then** p:=s[i]+p

 **else** p:=p+s[i];

 writeln (p);

**End**.

**Задача 5.**

**program** z4;

 **var** a, b, c :string; d, max:char;

 i,j, n,m, k,l : integer;

**begin**

readln(a);

 readln(b);

 n:=length(a);m:=length(b);

 k:=0;c:='';

 **for** i := 1 **to** n **do**

 **for** j:=1 **to** m **do**

 **if** a[i]=b[j]

 **then begin**

c:=c+b[j];

 k:=k+1;

 a[i]:='\*';

 b[j]:='!';

 **end**;

 **if** k=0 **then** writeln(-1)

 **else begin**

 **if** k>1 **then**

 **begin**

 **for** i:=1 **to** length(c)-1 **do**

 **begin**

max:=c[i];l:=i;

 **for** j:=i+1 **to** length(c) **do**

 **if** c[j]>max **then begin**

max:=c[j];

 l:=j;

 **end**;

 d:=c[l];

 c[l]:=c[i];

 c[i]:=d;

 **end**;

 **end**;

 writeln (c);

 **end**;

**End**.