

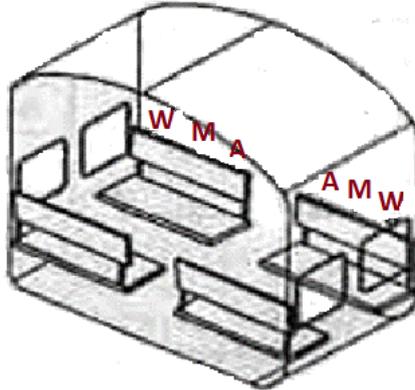
**Заключительный (очный) этап научно-образовательного соревнования
Олимпиады школьников «Шаг в будущее» по программированию «Профессор Лебедев»
(общеобразовательный предмет информатика), весна 2020 г.**

8 класс

Вариант 1

Задача 1

Наташа и **Олег** очень любят путешествовать. В основном они путешествуют по железной дороге. Однажды они ехали в вагоне, где были только сидячие места, и заинтересовались расположением сидений в своем купе. Купе выглядело примерно так



Тип места обозначается следующим образом:

- Сиденье у окна: **W**
- Среднее место: **M**
- Место у прохода: **A**

Нумерация мест в вагоне приведена на следующем рисунке

7	6	19	18	31	30	43	42	55	54	67	66	79	78	91	90	103	102
5	8	17	20	29	32	41	44	53	56	65	68	77	80	89	92	101	104
9	4	21	16	33	28	45	40	57	52	69	64	81	76	93	88	105	100
3	10	15	22	27	34	39	46	51	58	63	70	75	82	87	94	99	106
11	2	23	14	35	26	47	38	59	50	71	62	83	74	95	86	107	98
1	12	13	24	25	36	37	48	49	60	61	72	73	84	85	96	97	108

Ребятам стало интересно узнать номер и тип сиденья, расположенного напротив. Напишите программу, которая поможет ребятам решить эту задачу.

Входные данные

На вход подается одно целое число **N** ($1 \leq N \leq 108$), обозначающее номер места.

Выходные данные

Целое число, и большая латинская буква **W**, **M** или **A**, разделенные одним пробелом, которые будут соответствовать номеру и типу соседнего сиденья.

Примеры

Входные данные	Выходные данные
19	18 М
29	32 М

Тесты

Входные данные	Выходные данные
19	18 М
29	32 М
96	85 W
33	28 А
24	13 W
38	47 М
66	67 W
47	38 М
12	1 W
4	9 А

Решение

```
seats = ['W', 'M', 'A']
m = int(input())
temp = m - 1
coupe = temp // 12
temp %= 12
side = temp // 6
temp %= 6
part = (temp // 3) ^ side
temp %= 3
seat_ar = 12 * (coupe + 1) - (m - 12 * coupe) + 1
if side == part:
    print('{} {}'.format(seat_ar, seats[temp]))
else:
    print('{} {}'.format(seat_ar, seats[2 - temp]))
```

Задача 2

Вася наблюдает, как строители копают фундамент для нового дома. Землю они вывозят одинаковыми самосвалами. По каким-то своим причинам всю выбранную землю вывозят со стройки в тот же день, в который ее выбрали, то есть даже за одним кубометром все равно вечером приедет самосвал. Ещё одна странность, которую заметил Вася, что первый в день самосвал всегда наполняли максимум на половину.

Васе известно, сколько дней копали фундамент, какой объем земли вывозили каждый день и сколько кубометров помещается в самосвал. Вася любил считать всё, что можно, и в этот раз он решил считать самосвалы, но отвлекался на домашнюю работу и теперь боится, что посчитал не верно. Напишите программу, которая поможет Васе проверить общее количество самосвалов, которые вывозили грунт.

В первой строке на вход программы подаются два натуральных числа: количество дней t , в течение которых копали фундамент, $1 < t \leq 10^3$ и объем земли v , который вмещается в один самосвал, $1 < v \leq 10^2$. На следующей строке на вход программы поступает n целых неотрицательных чисел: a_1, a_2, \dots, a_n . a_i – сколько земли выбрали в i -й день, $1 < a_i \leq 10^5$

Выведите одно целое число – количество требуемых самосвалов.

Пример

Входные данные	Выходные данные
2 10 10 15	4

Комментарий к примеру. Строители работали два дня, в первый день они извлекли 10 кубометров грунта, и увезли двумя самосвалами по 5 кубов каждый, на следующий день тоже понадобилось 2 самосвала: первый вывез 5, второй 10 кубов. Итого потребовалось 4 самосвала.

Тесты

Входные данные	Выходные данные	Баллы
2 10 10 15	4	0
2 10 5 5	2	10
2 10 2 2	2	10
2 5 2 2	2	10
2 5 3 3	4	10
2 10 6 6	4	10
10 4 11 2 3 4 5 6 7 8 9 10	25	10
10 6 11 2 3 4 5 6 7 8 9 10	20	10
10 2 11 2 3 4 5 6 7 8 9 10	40	10
10 3 11 2 3 4 5 6 7 8 9 10	32	10
10 100 11 2 3 4 5 6 7 8 9 10	10	10

Решение

```
n, v0 = map(int, input().split())
days = list(map(int, input().split()))
res = 0
for v in days:
    res += int(((v + 3 * v0 / 2 - 0.5) // v0) if v > 0 else 0)
print(res)
```

Задача 3

Исполнитель получает на вход натуральное число X (не превышающее 10^6). По этому числу, точнее по его представлению в восьмеричной системе счисления, строится новое число Y по следующим правилам.

В восьмеричном представлении числа X предпоследняя цифра увеличивается на 1 (гарантируется, что в восьмеричном представлении X числа больше 2-х цифр). Например, $695_{10} = 1267_8 \rightarrow 1277_8 = 703_{10}$.

Если предпоследняя цифра 7, тогда предпоследняя цифра становится 0, а последняя изменяется по следующему принципу: четная увеличивается на 1, а нечетная уменьшается на 1. Например, последняя цифра нечетная $697_{10} = 1271_8 \rightarrow 1200_8 = 640_{10}$, последняя цифра четная $698_{10} = 1272_8 \rightarrow 1203_8 = 643_{10}$.

Введем понятие расстояния

$Oh = \text{Исходное_число} - \text{Полученное_число}$

Напишите программу, которая будет считать для скольких чисел из заданного интервала [A,B] расстояние Oh будет положительным.

На вход программы подаётся два целых числа A и B ($10 \leq A \leq B \leq 1\,000\,000$), записанных через пробел.

Программа должна вывести одно целое число - вычисленное значение.

Пример

Входные данные	Вывод	Примечание
694 698	3	$694 - 702 < 0$ ($1266_8 - 1276_8$) $695 - 703 < 0$ ($1267_8 - 1277_8$) $696 - 541 > 0$ ($1270_8 - 1201_8$) $697 - 640 > 0$ ($1271_8 - 1200_8$) $698 - 643 > 0$ ($1272_8 - 1203_8$)

Тесты

Ввод	Вывод
2356 2360	1
502 506	3
4087 4087	0
1914 1914	1
10 1000000	125000
999900 1000000	16
1000000 1000000	0
10 10	0
56 56	1
63 63	1

Решение

```

Program isp3_1;
var x,a,b,k:longint;
const d=8;
function digit(x:longint):longint;
var a,b,c:longint;
begin
  b := x mod d;
  a :=(x mod (d*d))div d;
  c := x div (d*d);
  a := a+1;
  if (a>=d)then
  begin
    a := 0;
    if(b mod 2=0) then b := b+1
    else b := b-1;
  end;
end;

```

```

    end;
    x:= c*d*d +a*d+ b;
    digit := x
end;
begin
k:=0;
readln(a,b);
for x:=a to b do
    if x-digit(x)>0 then
        k:=k+1;
writeln(k);
end.

```

Задача 4

Сообщение, которое передают по каналу связи, состоит из чисел, записанных в десятичной системе счисления. Каждое число состоит из шести знаков, а его двоичная запись оканчивается тремя нулями. При передаче сообщение было засорено посторонними шумами: числами, отличающимися от тех, что были в сообщении. Найдите изначальное количество чисел в сообщении.

Формат ввода

В строке вводится сначала целое число n – количество чисел в сообщении ($n \leq 1000$), затем n натуральных чисел, все числа отделены друг от друга одним или несколькими пробелами.

Формат вывода

Вывести одно целое число – количество достоверных сигналов в сообщении.

Пример

Входные данные	Выходные данные
3 100128 4356 234064	2

Тесты

Входные данные	Выходные данные
3 100128 4356 234064	2
4 234864 123048 230 1000	2
5 123048 547800 234000 123400 345864	5
6 100400 323464 100000 433864 64 32	4
1 100000	1
5 123064 345080 345 32 500800	3
2 40 111400	1
3 40 111400 11400	1
4 40 111400 11400 222800	2
5 40 111400 11400 222800 131072	3

Решение

```

program z83z92v3;

```

```

function good(num:integer):boolean;
var sum,k:integer;
begin
    sum:=0;

```

```

k:=0;
while num > 0 do
begin
sum:=sum+num mod 10;
num:=num div 10;
k:=k+1;
end;
good := (true) and (k=6);
end;

var x,m,n,i:integer;
begin
read(n);
m:=0;
for i:=1 to n do
begin
read(x);
if good(x) and (x mod 8 = 0) then
m:=m+1;
end;
writeln(m);
end.

```

Задача 5

Несколько агентов пересылают кодовые сообщения в Центр. Сообщение каждого агента представляет собой несколько слов, к каждому из которых приписан его идентификатор, состоящий из одной буквы. Сообщения записаны по очереди. Найдите сообщение, содержащее больше всего слов.

Формат ввода

В первой строке вводится сначала целое число n – количество слов ($n \leq 1000$), затем в n следующих строках записано по слову. Слова состоят только из строчных латинских букв.

Формат вывода

Вывести одно целое число – длину сообщения (количество слов в сообщении), содержащего больше всего слов.

Примеры

Входные данные	Выходные данные
4 bcd bcfd bcdfe abc	2
3 abd abdc bvd	1

Тесты

Входные данные	Выходные данные
4 bcd bcfd bcdfe abc	2
3 abd abdc bvd	1
5 b bb bbb bbbb bbbbb	5
7 abc abdc f g gh dfh cfh	3
5 a ab c d e	1
8 abv vbav fgh dfge fhn fth aav aab	2
1 asdvbn	1
10 abc abc abc df ad acf abc	3

abc d abc	
5 a b bb bbb bbbb	4
3 abc abc d	2

Решение

```
program z85z94v1_1203;
```

```
var
```

```
  n,i,x,len,maxlen:integer;
```

```
  s,ps:string;
```

```
begin
```

```
  readln(n);
```

```
  maxlen:=0;
```

```
  len:=1;
```

```
  ps:='';
```

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

```
  begin
```

```
    readln(s);
```

```
    if copy(ps,length(ps),1) = copy(s,length(s),1) then
```

```
      begin
```

```
        len:=len+1;
```

```
      end
```

```
    else
```

```
      begin
```

```
        if len>maxlen then
```

```
          maxlen:=len;
```

```
          len:=1;
```

```
        end;
```

```
      ps:=s;
```

```
    end;
```

```
  if len>maxlen then
```

```
    maxlen:=len;
```

```
  writeln(maxlen);
```

```
end.
```

Ситуационная задача 1.

Павел ведет дневник, наблюдая за погодой. Каждый день он записывает температуру. Требуется определить, в какой день во время периода наблюдений была наименьшая температура и сколько дней отмечалась температура выше порогового значения

На вход программе в первой строке подаются натуральное число N – количество дней, в течение которых велось наблюдение ($N \leq 20$) и вещественное число T – пороговое значение температуры.

Далее в N строках подается на вход по вещественному числу – t_i : Температура в i -й день.

Вывести два целых числа – в первой строке вывести номер дня, в который отмечалась наименьшая температура, во второй строке вывести, сколько дней отмечалась температура выше порогового значения. Если наименьшая температура наблюдалась в несколько разных дней, вывести наименьший номер дня.

Пример

Ввод	Вывод
3 5	1
5	2
6	
7	

Тесты

Ввод	Вывод
1 1	1
1.1	1
5 0	1
0	0
0	
0	
0.0	
5 0	5
-1	0
-1.1	
-2.1	
-0.3	
-5	
10 2	1
1.5	7
1.7	
1.9	
2.1	
2.4	
2.3	
2.5	
4.0	
4.5	
5.1	
8 -2	7
-3	1

-2.4 -4 -4.5 -6 -5.6 -7.2 -1	
3 0 60 55 60	2 3
5 40 39 38 40 41 42	2 2
1 1 1	1 0
3 -1 -1 -1.1 -0.9	2 1
4 -1.5 -1 -1.1 -2 -1.9	3 2

Решение

program z801_1103;

var

n,tj,j,k:integer;

t,ti,tmax:real;

begin

 readln(n,t);

 k:=0;

 tj:=1;

 readln(tmax);

if tmax>t **then**

 k:=k+1;

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

begin

 readln(ti);

if ti<tmax **then**

begin

 tj:=j;

 tmax:=ti;

end;

if ti>t **then**

```

    k:=k+1;
end;
writeln(tj);
writeln(k);
end.

```

Ситуационная задача 2.

Робот может выполнять команды «Поиск дефектов», «Движение», «Подготовка» и «Ремонт». Из-за конструктивных особенностей на робота наложены некоторые ограничения. Два раза подряд можно выполнить только команду «Движение». Команда «Ремонт» может быть выполнена только на следующем шаге после команды «Подготовка».

Напишите программу, которая определит, сколько существует выполнимых последовательностей команд длиной n , если до начала выполнения программы робот выполнил команду «Подготовка».

На вход программе подается натуральное число n ($n \leq 15$) – количество команд.

Вывести целое число – количество выполнимых последовательностей команд длиной n .

Пример

Ввод	Вывод
2	8

Тесты

Ввод	Вывод
1	3
3	22
10	24960
12	186304
15	3799168

Решение

```

program z8102_1103;
var
  i,n,a,b,c,d,pa,pb,pc,pd:integer;
begin
  readln(n);
  a:=0; //Поиск
  b:=0; //Движение
  c:=1; //Подготовка
  d:=0; //Ремонт
  for i:=1 to n do
    begin
      pa:=a; pb:=b; pc:=c; pd:=d;
      a:=pb+pa+pd+pc;
      b:=pa+pc+pd;
      c:=pa+pb+pd;
      d:=pc;
    end;
  writeln(a+b+c+d);
end.

```

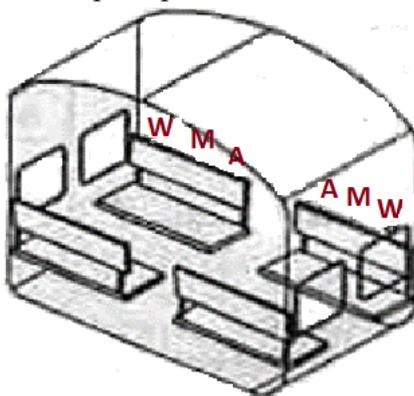
**Заключительный (очный) этап научно-образовательного соревнования
Олимпиады школьников «Шаг в будущее» по программированию «Профессор Лебедев»
(общеобразовательный предмет информатика), весна 2020 г.**

8 класс

Вариант 2

Задача 1

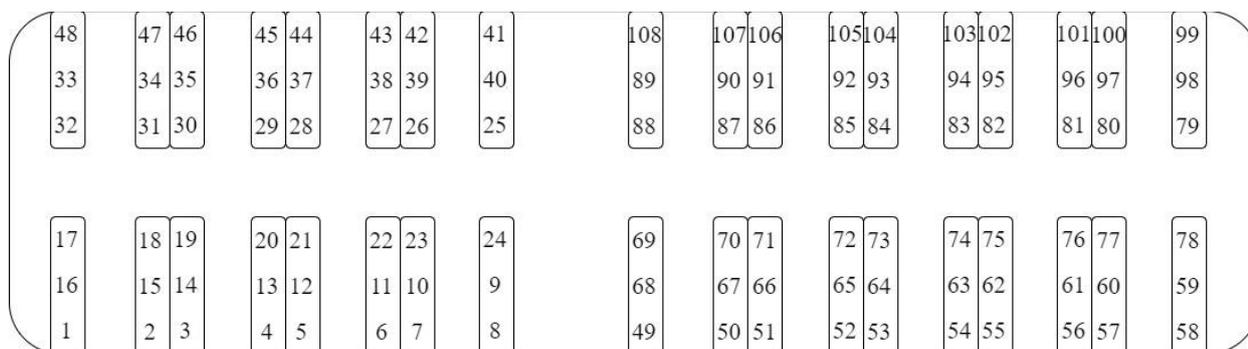
Миша и **Маша** очень любят путешествовать. В основном они путешествуют по железной дороге. Однажды они ехали в вагоне, где были только сидячие места, и заинтересовались расположением сидений в своем купе. Купе выглядело примерно так



Тип места обозначается следующим образом:

- Сиденье у окна: **W**
- Среднее место: **M**
- Место у прохода: **A**

Нумерация мест в вагоне приведена на следующем рисунке



Ребятам стало интересно узнать номер и тип сиденья, расположенного напротив. Напишите программу, которая поможет ребятам решить эту задачу.

Входные данные

На вход подается одно целое число N ($1 \leq N \leq 108$), обозначающее номер места.

Выходные данные

Целое число, и большая латинская буква **W**, **M** или **A**, разделенные одним пробелом, которые будут соответствовать номеру и типу сиденья, расположенного напротив.

Примеры

Входные данные	Выходные данные
19	20 A
96	95 A

Тесты

Входные данные	Выходные данные
19	20 A
96	95 A
29	30 A
33	34 M
24	23 A
38	37 M
66	65 A
47	48 W
12	11 M
4	3 W

Решение

```
n = int(input())
seat = ""
if n <= 48:
    r = (n - 1) // 8
else:
    r = (n - 1) // 10
if r == 0 or r == 5:
    seat = 'W'
elif r == 1 or r == 4:
    seat = 'M'
else:
    seat = 'A'
print(n + (-1) ** (n % 2 + 1), seat)
```

Задача 2

Петя наблюдает, как строители копают фундамент для нового дома. Землю они вывозят одинаковыми самосвалами. По каким-то своим причинам всю выбранную землю вывозят со стройки в тот же день, в который ее выбрали, то есть даже за одним кубометром все равно вечером приедет самосвал.

Пете известно, сколько времени копали фундамент, какой объем земли вывозили каждый день и сколько кубометров помещается в самосвал. Он задался вопросом: насколько меньше рейсов понадобилось бы совершить, если бы можно было вывозить землю не в тот же самый день, в который ее извлекли?

Напишите программу, которая решит эту задачу за Петю.

В первой строке на вход программы подаются два натуральных числа: количество дней t , в течение которых копали фундамент, $1 < t \leq 10^3$ и объем земли v , который вмещается в один самосвал, $1 < v \leq 10^2$. На следующей строке на вход программы поступает n натуральных чисел: a_1, a_2, \dots, a_n . a_i – сколько земли выбрали в i -й день, $1 < a_i \leq 10^5$.

Выведите одно целое число – разницу между количеством проделанных рейсов и минимальным количеством рейсов, за которые можно вывезти этот объем земли.

Пример

Входные данные	Выходные данные
2 4 6 5	1

Комментарий к примеру. Строители работали два дня, в первый день они извлекли 6 кубометров грунта, во второй – 5. Самосвал принимает 4 кубометра грунта, следовательно, понадобится 4 рейса. Можно было обойтись за три рейса, следовательно, разница – 1.

Тесты

Входные данные	Выходные данные	Баллы
2 4 6 5	1	0
4 5 2 2 2 2	2	10
5 5 2 2 2 2 2	3	10
6 5 2 2 2 2 2 2	3	10
5 5 5 5 5 5 5	0	10
5 5 5 5 4 5 5	0	10
5 5 6 2 3 4 5	2	10
10 5 11 12 13 14 15 16 17 18 19 20	4	10
10 6 11 12 13 14 15 16 17 18 19 20	4	10
10 7 11 12 13 14 15 16 17 18 19 20	3	10
5 100 100000 99999 99900 99899 99901	1	10

Решение

```
n, v0 = map(int, input().split())
days = list(map(int, input().split()))
res0 = total_v = 0
for v in days:
    res0 += (v + v0 - 1) // v0
    total_v += v
res1 = (total_v + v0 - 1) // v0
print(res0 - res1)
```

Задача 3

Исполнитель получает на вход натуральное число X (не превышающее 10^6). По этому числу, точнее по его представлению в шестеричной системе счисления, строится новое число Y по следующим правилам.

В шестеричном представлении числа X предпоследняя цифра увеличивается на 1 (гарантируется, что в шестеричном представлении X числа больше 2-х цифр). Например, $749_{10} = 3245_6 \rightarrow 3255_6 = 755_{10}$.

Если предпоследняя цифра 5, тогда предпоследняя цифра становится 0, а последняя изменяется по следующему принципу: четная увеличивается на 1, а нечетная уменьшается на 1. Например, последняя цифра нечетная $751_{10} = 3251_6 \rightarrow 3200_6 = 720_{10}$, последняя цифра четная $752_{10} = 3252_6 \rightarrow 3203_6 = 723_{10}$.

Введем понятие расстояния

$Oh = \text{Исходное_число} - \text{Полученное_число}$

Напишите программу, которая будет считать для скольких чисел из заданного интервала $[A, B]$ расстояние Oh будет отрицательным.

На вход программы подаётся два целых числа A и B ($10 \leq A \leq B \leq 1\,000\,000$), записанных через пробел.

Программа должна вывести одно целое число - вычисленное значение.

Пример

Входные данные	Вывод	Примечание
748 752	2	$748 - 754 < 0$ ($3244_6 - 3254_6$) $749 - 755 < 0$ ($3245_6 - 3255_6$) $750 - 721 > 0$ ($3250_6 - 3201_6$) $751 - 720 > 0$ ($3251_6 - 3200_6$) $752 - 723 > 0$ ($3252_6 - 3203_6$)

Тесты

Ввод	Вывод
1034 1038	4
208 212	2
4087 4087	1
1940 1940	0
10 1000000	833329
999900 1000000	89
1000000 1000000	1
10 10	1
55 55	1
56 56	0

Решение

```
Program isp4_1;  
var x,a,b,k:longint;  
const d=8;  
function digit(x:longint):longint;  
var a,b,c:longint;  
begin  
  b := x mod d;  
  a :=(x mod (d*d))div d;
```

```

c := x div (d*d);
a := a+1;
if (a>=d)then
  begin
    a := 0;
    if(b mod 2=0) then b := b+1
    else b := b-1;
  end;
x:= c*d*d +a*d+ b;
digit := x
end;
begin
k:=0;
readln(a,b);
for x:=a to b do
  if x-digit(x)<0 then
    k:=k+1;
writeln(k);
end.

```

Задача 4

Зонд передает данные с орбиты Юпитера во время сильной магнитной бури. Информация передается по каналу связи в виде пакетов. Каждый пакет представляет собой целое положительное число в двоичной системе счисления. Для обнаружения помех последний разряд в пакете подбирают таким образом, чтобы количество единиц в разрядах пакета было четным. В каждом пакете на практике никогда не искажается больше одного разряда.

Напишите программу, которая по распечатке пакетов, записанных в десятичной системе счисления, найдет самое большое значение, прошедшее без искажений. Известно, что как минимум один пакет прошел без искажений.

Формат ввода

В строке вводится сначала целое число n – количество пакетов ($n \leq 1000$), затем n натуральных чисел, все числа отделены друг от друга одним или несколькими пробелами.

Формат вывода

Вывести одно целое число – самое большое значение, прошедшее без искажений.

Пример

Входные данные	Выходные данные
4 1025 496 882 1056	1056

Тесты

Входные данные	Выходные данные
4 1025 496 882 1056	1056
5 1025 4 5 6 7	1025
5 512 234 2051 888 2049	2049
10 1 1 1 1 1 4097 1 1 1 1	4097
8 2456 23445 4086 511 4096 256 23546 854	4086
6 1001 1001 513 1001 1001 1001	513
8 2466 23545 4186 521 4036 456 25547 864	864
6 1101 1011 513 10010 10111 10010	1011
10 12 13 14 15 16 4097 17 18 19 10	4097
10 12 13 14 15 16 4096 17 18 19 10	18

Решение

```
program z83z92v3;
```

```
function good(num:integer):boolean;
```

```
var sum:integer;
```

```
begin
```

```
  sum:=0;
```

```
  while num > 0 do
```

```
  begin
```

```
    sum:=sum+num mod 2;
```

```
    num:=num div 2;
```

```
  end;
```

```
good:=sum mod 2 = 0;
```

```
end;
```

```
var x,m,n,i,max:integer;
```

```
begin
```

```
  read(n);
```

```
  max:=0;
```

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

```
  begin
```

```
    read(x);
```

```
    if good(x) and (x>max) then
```

```
      max:=x;
```

```
  end;
```

```
  writeln(max);
```

```
end.
```

Задача 5

Вася придумывает пароль для каждой новой учетной записи, которую он заводит на каком-то из своих устройств. Время от времени он изменяет пароль, дописывая к нему новые символы. Предыдущий пароль никогда не будет началом пароля для новой учетной записи Васи. После того, как Вася заводит новую учетную запись, он перестает менять пароль на старой.

Зная все Васины пароли в хронологическом порядке, напишите программу, которая найдет, какое наибольшее количество раз Вася менял пароль для одной учетной записи.

Формат ввода

В первой строке вводится сначала целое число n – количество слов ($n \leq 1000$), затем в n следующих строках записано по слову. Слова состоят только из строчных латинских букв.

Формат вывода

Вывести одно целое число – какое наибольшее количество раз Вася менял пароль для одной учетной записи.

Примеры

Входные данные	Выходные данные
3 abd abdc bvd	1

4 bcd bcdf bcdfe abc	2
----------------------------------	---

Тесты

Входные данные	Выходные данные
3 abd abdc bvd	1
4 bcd bcdf bcdfe abc	2
5 a aa aaa aaaa aaaaa	4
7 a b cd df fg fg ab	0
8 a b c d e f fg fgh	2
5 a ab c d e	1
1 asdvbn	0
8 a	

ab c cd cde f fg h	
8 a ab abc abcd cde f fg h	3
8 f fg h a ab abc abcd abcde	4

Решение

```
program z85z94v1_1203;
```

```
var
```

```
n,i,x,len,maxlen:integer;
```

```
s,ps:string;
```

```
begin
```

```
  readln(n);
```

```
  maxlen:=0;
```

```
  len:=0;
```

```
  ps:='';
```

```
  for i:=1 to n do
```

```
  begin
```

```
    readln(s);
```

```
    if (pos(ps,s)=1) and (length(ps)<length(s)) then
```

```
      begin
```

```
        len:=len+1;
```

```
      end
```

```
    else
```

```
      begin
```

```
        if len>maxlen then
```

```
          maxlen:=len;
```

```
        len:=0;
```

```
      end;
```

```

ps:=s;
end;
if len>maxlen then
maxlen:=len;
writeln(maxlen);
end.

```

Ситуационная задача 1.

Миша ведет дневник, отслеживая, сколько шагов в день он проходит пешком. Каждый день он записывает количество шагов. Требуется определить, в какой день во время периода наблюдений он прошел наибольшую дистанцию и прошел ли он столько, сколько планировал, за все время.

На вход программе в первой строке подаются натуральное число N – количество дней, в течение которых велось наблюдение ($N \leq 20$) и натуральное число X – запланированное количество шагов.

Далее в N строках подается на вход по целому положительному числу – x_i : количество шагов, пройденных в i -й день.

Вывести два целых числа – в первой строке вывести номер дня, в который Миша прошел больше всего шагов, во второй строке вывести 1, если он прошел не менее запланированного числа шагов и 0 в обратном случае. Если Вася прошел одинаковое количество шагов в несколько разных дней, вывести наименьший номер дня.

Пример

Ввод	Вывод
3 10000 5100 6500 4200	2 1

Тесты

Ввод	Вывод
1 5000 4900	1 0
2 10 6 7	2 1
3 15 5 3 6	3 0
10 50000 4000 4500 5000 5200 7000 8900 8765 2345 9023	9 1

1298	
3 3	1
1	1
1	
1	

Решение

program z801_1103;

var

n,x,j,k,x,xi,xmax:integer;

begin

 readln(n,x);

 k:=0;

 readln(xmax);

 xj:=1;

 k:=xmax;

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

begin

 readln(xi);

if xi>xmax **then**

begin

 xj:=j;

 xmax:=xi;

end;

 k:=k+xi;

end;

 writeln(xj);

if k>=x **then**

 writeln(1)

else

 writeln(0)

end.

Ситуационная задача 2.

Исследовательский аппарат на поверхности Марса может выполнять команды «Фотографирование», «Пробное бурение», «Взятие образцов грунта», «Анализ атмосферы». Из-за конструктивных особенностей на аппарат наложен ряд ограничений. Нельзя выполнять команду «Анализ атмосферы» после команды «Пробное бурение». Нельзя выполнить команду «Фотографирование» после команды «Взятие образцов грунта». Команду «Взятие образцов грунта» можно выполнять только следующей после команды «Пробное бурение». Никакую команду, кроме команды «Взятие образцов грунта», нельзя выполнить подряд дважды.

Напишите программу, которая определит, сколько существует выполнимых последовательностей команд длиной n, если до начала выполнения программы аппарат выполнил команду «Пробное бурение».

На вход программе подается натуральное число n ($n \leq 15$) – количество команд.

Вывести целое число – количество выполнимых последовательностей команд длиной n.

Пример

Ввод	Вывод
2	5

Тесты

Ввод	Вывод
1	2
3	11
5	56
12	16212
15	183922

Решение

program z8102_1103;

var

i,n,a,b,c,d,pa,pb,pc,pd:integer;

begin

readln(n);

a:=0; //Фото

b:=1; //Бурение

c:=0; //Грунт

d:=0; //Атмосфера

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

begin

pa:=a; pb:=b; pc:=c; pd:=d;

a:=pb+pd;

b:=pa+pc+pd;

c:=pb+pc;

d:=pa+pc;

end;

writeln(a+b+c+d);

end.