

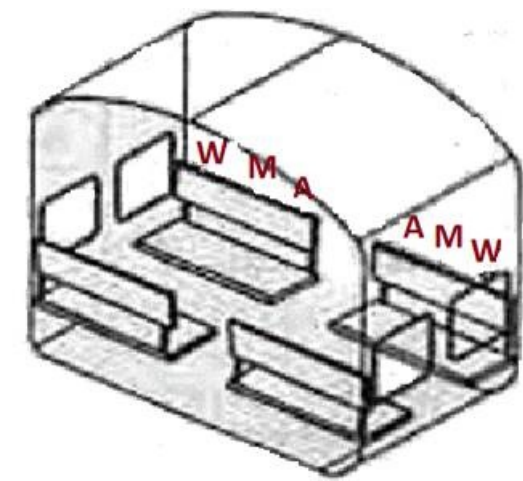
Плеханов Владимир Александрович

Балл: 111

Задача №1

Условие

Витя 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

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

Примеры

Входные данные	Выходные данные
19	18 W

Исходный код

```
#include <iostream>
using namespace std;
int main(){
    int a = 0, b;
    char q;
    cin >> a;

    if (a % 6 == 1){
        q = 'W';
    }
    else{
        if (a % 6 == 2){
            q = 'M';
        }
        else{
            if (a % 6 == 3){
                q = 'A';
            }
            else{
                if (a % 6 == 4){
                    q = 'A';
                }
                else{
                    if (a % 6 == 5){
                        q = 'M';
                    }
                    else{

                        }
                    }
                }
            }
        }
    }
    if (a % 12 == 1){
        b = a + 11;
    }
    else{
        if (a % 12 == 2){
            b = a + 9;
        }
        else{
            if (a % 12 == 3){
                b = a + 7;
            }
            else{
                if (a % 12 == 4){
                    b = a + 5;
                }
                else{
                    if (a % 12 == 5){
                        b = a + 3;
                    }
                    else{
                        if (a % 12 == 6){
                            b = a + 1;
                        }
                        else{
                            if (a % 12 == 7){
                                b = a - 1;
                            }
                            else{
                                if (a % 12 == 8){
                                    b = a - 3;
                                }
                                else{
                                    if (a % 12 == 9){
                                        b = a - 5;
                                    }
                                    else{
                                        if (a % 12 == 10){
                                            b = a - 7;
                                        }
                                        else{
                                            if (a % 12 == 11){
                                                b = a - 9;
                                            }
                                            else{
                                                b = a - 11;
                                            }
                                        }
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    }
```

```

    }
    }
    }
    }
    }
    }
    }

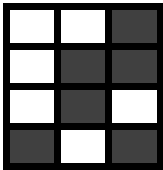
cout << b << ' ' << q;
}

```

Задача №2

Условие

Петя недавно узнал, что для записи чисел достаточно использовать только нолики и единички. Числа получаются длиннее, зато их можно изображать, закрашивая клеточки тетради, используя только карандаш или только ручку. Петя решил, что он будет закрашивать клеточку, когда встречается цифра один, и оставлять пустой, если цифра ноль. Например, вот такой рисунок Пети



означает число $001\ 011\ 010\ 101_2 = 512 + 128 + 64 + 16 + 4 + 1 = 725$.

Петя считает, что каждый рисунок должен обладать разнообразностью, т.е. не должно быть рисунков, в которых сильно преобладает какой-то один цвет. Лучше, когда пустых и закрашенных клеточек приблизительно поровну. Теперь Петя задался вопросом, сколько разных чисел можно изобразить такими картинками.

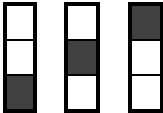
На вход программе подаются три натуральных числа ширина и высота картинки в клеточках, а также количество клеточек, которые должны быть закрашены, чтобы картинки обладали разнообразностью. Общее количество клеточек в изображении не превышает 30, количество клеточек, которые должны быть закрашены меньше общего количества клеточек.

Выведите, сколько разных чисел можно изобразить заданным образом.

Пример

Входные данные	Выходные данные
1 3 1	3

Пояснение к примеру. С одной закрашенной клеточкой разных рисунков размера 3 x 1 может быть только три:



Исходный код

```

#include <iostream>
using namespace std;
int main(){
    long long a, b, c, l=1;
    cin >> a >> b >> c;
    if (a*b - c >= c){
        for (int i = 1; i <= a*b; i++){
            if (a*b - c < i){
                l = l*i;
            }
        }
    }
}

```

```

for (int i = 1; i <= c; i++){
    l = l / i;
}
}
else{
    for (int i = 1; i <= a*b; i++){
        if (c < i){
            l = l*i;
        }
    }
    for (int i = 1; i <= a*b - c; i++){
        l = l / i;
    }
}
cout << l;
}

```

Задача №3

Условие

Исполнитель Цифропоглотитель получает на вход натуральное число X (не превышающее 10^6). По этому числу строится трёхзначное восьмеричное число Y по следующим правилам.

1. Первая цифра числа Y – остаток от деления X на 2.
2. Вторая цифра числа Y – остаток от деления X на 3.
3. Третья цифра числа Y – остаток от деления X на 5.

Пример. Исходное число: 35. Остаток от деления на 2 равен 1; остаток от деления на 3 равен 2; остаток от деления на 5 равен 0. Результат работы автомата: 120_8 .

(Замечание: $120_8 = 80_{10}$)

Напишите программу, которая будет находить наибольшее десятичное число, которое можно получить на заданном интервале $[A, B]$, и наибольшее число, на котором оно достигается (например, число 80 можно получить из чисел 5 и 35, наибольшее из них 35, оно и пойдет в ответ).

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

Программа должна вывести одно целое число - вычисленное значение.

Входные данные	Вывод	Примечание
34 40	80 35	<p>34 -> 12</p> <p>35 -> 80</p> <p>36 -> 1</p> <p>37 -> 74</p> <p>38 -> 19</p> <p>39 -> 68</p> <p>40 -> 8</p> <p>Наибольшее значение, получаемое на заданном интервале 80. Оно получается из числа 35.</p>

Исходный код

```

#include <iostream>
using namespace std;
int main(){
    int a, b, c=0, r;
    cin >> a >> b;

```

```

for (int i = b; i >= a; i--){
    if ((i % 2) * 64 + (i % 3) * 8 + (i % 5) > c){
        c = (i % 2) * 64 + (i % 3) * 8 + (i % 5);
        r = i;
    }
}
cout << r << ' ' << c;
}

```

Задача №4

Условие

Станция связи принимает сообщения от нескольких зондов. Каждый зонд посылает сообщение в виде нескольких идущих подряд пакетов фиксированной длины. Пакет состоит из символов. Длина пакета для каждого зонда своя. Из-за ошибки в программе станция связи обрабатывает каждый пакет в течение одной секунды.

Напишите программу, которая найдет, сколько секунд станция обрабатывала самую долгую передачу.

Формат ввода

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

Формат вывода

Вывести одно целое число — сколько секунд станция обрабатывала самую долгую передачу.

Примеры

Входные данные	Выходные данные
3 abc aab abad	2
4 abmn abdv abcd abcfde	3

Исходный код

```

#include <iostream>
#include <vector>
#include <string>
using namespace std;
int main(){
    int n, l = 0, m = 0;
    cin >> n;
    vector <string> r(n);
    vector <int> s;
    vector <int> t;
    for (int i = 0; i < n; i++){
        cin >> r[i];
        for (int j = 0; j < t.size(); j++){
            if (t[j] == r[i].size()){
                m++;
                s[j]++;
                break;
            }
        }
    }
}

```

```

if (m==0){
    t.push_back(r[i].size());
    s.push_back(1);
}
m = 0;
}
for (int i = 0; i < s.size(); i++){
    if (s[i]>1){
        l = s[i];
    }
}
cout << l;
}

```

Задача №5

Условие

Миша собирает со своей команды данные о количестве тренировок. Игроки сообщают о тренировках, вписывая в форме опроса в одном окне сначала количество сыгранных игр, а потом – свой номер. Номер игрока представляет собой двузначное число. Собрав данные, Миша увидел, что все пробелы между количеством игр и номером игрока были удалены. Напишите программу, которая поможет Мише найти игрока с наибольшим количеством тренировок.

Формат ввода

В строке вводится сначала целое число n – количество записей ($n \leq 1000$), затем n натуральных чисел, каждое из которых не менее 100. Все числа отделены друг от друга одним или несколькими пробелами.

Формат вывода

Вывести одно целое число – номер игрока, сыгравшего больше всего игр.

Пример

Входные данные	Выходные данные
4 1125 496 882 1056	25

Исходный код

```

#include <iostream>
#include <vector>
#include <string>
using namespace std;
int main(){
    int n, r, s;
    cin >> n;
    vector<int> a(n);
    vector<int> b(n);
    for (int i = 0; i < n; i++){
        cin >> r;
        a[i] = r % 100;
        b[i] = r / 100;
    }
    r = 0;
    for (int i = 0; i < n; i++){
        if (r < b[i]){
            r = b[i];
            s = a[i];
        }
    }
    cout << s;
}

```

Задача №6

Условие

В цепи постоянного тока под напряжением U надо получить силу тока не больше, чем I , включив нагрузку. Получить это напряжение можно, включив в цепь один из нескольких предложенных резисторов. Для каждого резистора известны его сопротивление и цена. Найти резистор, включить который выйдет дешевле всего. Гарантируется, что хотя бы один подходящий резистор есть.

На вход программе в первой строке подаются натуральное число N – количество резисторов ($N \leq 20$) и два вещественных положительных числа: U – требуемое напряжение, I – сила тока в цепи

Далее в N строках подается на вход пара вещественных неотрицательных чисел – r_i и p_i : сопротивление i -го резистора и цена за него соответственно.

Вывести одно натуральное число – номер самого дешевого подходящего резистора.

Количество теплоты рассчитывается по формуле

$$U = IR$$

где U – напряжение, I – сила тока, R – сопротивление.

Пример

Ввод	Вывод
2 380 0.001	1
1000000 1005.9	
1000 1	

Исходный код

```
#include <iostream>
#include <vector>
#include <string>
using namespace std;
int main(){
    double n, I, U, R, r, t, l = 999999999, s;
    cin >> n >> U >> I;
    R = U / I;
    for (int i = 0; i < n; i++){
        cin >> r >> t;
        if (r >= R){
            if (l > t){
                l = t;
                s = i;
            }
        }
    }
    cout << s+1;
    return 0;
}
```

Задача №7

Условие

Автоматика, управляющая грузовым лифтом, оперирует только четырьмя командами: «Открыть двери», «Закрыть двери», «Поехать на второй этаж», «Поехать на первый этаж». Из-за недоработок в системе нельзя два раза подряд выполнить команду «Закрыть двери». Остальные попытки обрабатываются системой корректно.

Сколько существует последовательностей команд длиной n , не приводящих к ошибке и не содержащих двух последовательных перемещений между этажами, если лифт в начальный момент времени стоял на втором этаже?

На вход программе подается натуральное число n ($N \leq 20$) – количество команд.

Вывести целое число – количество последовательностей команд длиной n , не приводящих к ошибке и не содержащих двух одинаковых команд подряд.

Пример

Ввод	Вывод
2	7

Исходный код

```
#include <iostream>
#include <vector>
#include <string>
using namespace std;
int main(){
    int n, j=1, p=1, d1=0, d2=0, x = 2, j1, p1, x1;
    cin >> n;
    for (int i = 0; i < n; i++){
        j1 = j;
        p1 = p;
        x1 = x;
        x = x + x - p + j + p + j + p;
        j = x1;
        p = x1 - p;
        d1 = p1 + j1;
        d2 = p1 + j1;
    }
    cout << x1;
    return 0;
}
```