Цителова Елизавета Дмитриевна

Балл: 101

Задача №1

Условие

Обычно Петя добирается до школы на автобусе. Иногда приходится ехать в битком набитом автобусе, иногда наоборот, в почти пустом. Петя заметил, что он не один, кто предпочитает не сидеть, а стоять. По его подсчётам предпочитает стоять каждый пятый пассажир. Петя составил таблицу, которая показывает, при какой заполненности автобуса сколько пассажиров предпочитают стоять.

Общее количество пассажиров в автобусе	Количество пассажиров, предпочитающих сидеть	Количество пассажиров, предпочитающих стоять
1	1	0
2	2	0
3	3	0
4	3	1
5	4	1
6	5	1
7	6	1
8	7	1
9	7	2
…	…	…

Четыре пассажира, уже почти пять, поэтому среди них найдётся один, кто предпочтёт стоять.

Петя считает сиденье бесполезным, если на нём никто не сидит. Общей бесполезностью Петя называет суммарное количество, сколько остановок пустовало каждое сидение. Например, если от начала поездки до конца, нужно ехать 3 остановки и каждый раз пустовало каких-то 4 сидения, то общая бесполезность за время поездки равнялось 12. Каждый раз, когда Петя едет в школу, он считает общую бесполезность. Напишите программу, которая делает это вместо Пети.

Сначала на вход программы подаются два числа: количество сидячих мест s, $1 < s \leq 10^3$ и количество остановок n, $1 < n \leq 10^5$ (на первой пассажиры могут только заходить, на n-ой все выходят). Далее на вход программы поступает ещё n-1 число: $a_1, a_2, \ldots, a_{n-1}$. a_i – это на сколько пассажиров увеличилось общее количество пассажиров в автобусе после i-ой остановки. $a_i < 0$ означает, что на i-ой остановке больше пассажиров вышло, чем вошло. Гарантируется, что число пассажиров в автобусе никогда не оказывается отрицательным и не оказывается больше 1000.

Выведите одно натуральное число – общую бесполезность сидений в автобусе.

Пример

Входные данные	Выходные данные
10 3 8 -2	8

Комментарий к примеру. На первой остановке в автобус вошли 8 человек, семеро из них успешно сели, 3 сидения остались бесполезными. На следующей остановке двое вышли, осталось 6 человек, из которых сидели пятеро, бесполезными оказалось 5 сидений. На третьей остановке все вышли, т.к. это была конечная. Общая бесполезность сидений оказалась равна восьми.

Исходный код

```
#include <iostream>
```

```
using namespace std;

int main()
{
    int s,n;
    cin>>s>>n;
    int a[n-1];
    for(int i=0;i<n-1;i++)
        cin>>a[i];
    int ans=0;
    int pe=0;
    for(int i=0;i<n-1;i++)
    {
        pe+=a[i];
        int tmp=pe/5;
        if(pe%5==4)
            tmp++;
        ans+=(s-(pe-tmp));
    }
    cout<<ans;
    return 0;
}
```

Задача №2

Условие

Вася любит придумывать миры для настольных игр. При рисовании политической карты очередного мира он задается вопросом: можно ли раскрасить эту карту с помощью заранее известного количества цветов так, чтобы никакие два соседних государства не оказались окрашены одним цветом?

Напишите программу, которая решит эту задачу за Васю.

Сначала на вход программы подаются два натуральных числа: количество цветов n, в которые можно красить карту, $1 < n \leq 5$, и количество стран на карте k, $1 < k \leq 8$. Далее на вход программы подаются k строк по k чисел – матрица смежности стран. j-ое число i-ой равно единице, если между i-ой и j-ой странами есть общая граница и ноль, если общей границы у них нет. Считается, что сама с собой страна не граничит.

Выведите одно целое число – сколькими способами можно раскрасить карту.

Пример

Входные данные	Выходные данные
2 3 0 1 0 1 0 1 0 1 0	2

Комментарий к примеру.

В распоряжении Васи два цвета, на его карте три страны. Страны №1 и №2 граничат только со страной №2. Соответственно, можно покрасить страны №1 и №3 в цвет №1, а страну №2 в цвет №2, можно наоборот, покрасить страны №1 и №3 в цвет №2, а страну №2 – в цвет №1.

Исходный код

```
#include <iostream>

using namespace std;
bool us[8];
//int h=0;
 int usi[8];
 int m[8][8];
/*void fu(int v,int n,int k)
{
```

```
    if(us[v]==1)
        return;
    us[v]=1;
    if(usi[v]>=n-1)
        h++;
    for(int i=0;i<k;i++)
        if(m[v][i]==1)
            fu( i,n,k);
}*/
int main()
{
    int n;
    int k;
    cin>>n>>k;


    for(int i=0;i<8;i++)
        usi[i]=0;
    for(int i=0;i<k;i++)
    for(int j=0;j<k;j++){
            cin>>m[i][j];
            if(m[i][j]==1)
            usi[i]++;
    }
    for(int i=0;i,k;i++)
    {
        if(usi[i]>=n-1)
        {
            int h=1;
            for(int j=0;j<k;j++)
                if(m[i][j]==1&&usi[j]>=n-1)
            {
                h++;
            }
            if(h>=n&&h!=1)
            {
                cout<<0;
                return 0;
            }

        }
    }
  /*  for(int i=0;i<k;i++)
    {
        if(us[i]==0&&usi[i]>=n)
        {
            h=0;
            fu(i,n,k);
            if(h>=n){
                cout<<0;
                return 0;
        }}
    }*/
    int y=1;
    for(int i=1;i<=n;i++)
        y*=i;
    cout<<y;
    return 0;
}
```

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 считать сколько однозначных десятичных чисел получится на

заданном интервале [A,B].

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
31 50	5	$36 \rightarrow 1_8 = 1_{10}$ $40 \rightarrow 10_8 = 8_{10}$ $42 \rightarrow 2_8 = 2_{10}$ $46 \rightarrow 11_8 = 9_{10}$ $48 \rightarrow 3_8 = 3_{10}$

Исходный код

```
#include <iostream>

using namespace std;

int main()
{
    int a,b;
    cin>>a>>b;
    int ans=0;
    for(int i=a;i<=b;i++)
    {
        if(i%2==0)
        {
            if(i%3==1&&i%5<2)
                ans++;
            if(i%3==0)
                ans++;
        }
    }
    cout<<ans;
    return 0;
}
```

Задача №4

Условие

Термодатчик измеряет температуру объекта раз в секунду и пересылает результат измерений в шестнадцатеричной системе счисления. Так как передача информации идет с сильными помехами, два последних разряда в каждом измерении заменяются нулями. Если передача данных шла с помехами, то эти нули заменяются на случайную цифру. Надежным считается значение, которое в шестнадцатеричной системе заканчивается двумя нулями. В распоряжении оператора находится распечатка результатов измерений в десятичной системе счисления. Напишите программу, которая найдет наивысшее надежное значение.

Формат ввода

В строке вводится сначала целое число ***n*** – количество секунд, в течение которых шла запись ($n \leq 1000$), затем ***n*** натуральных чисел, все числа отделены друг от друга одним или несколькими пробелами

Формат вывода

Вывести одно целое число – наивысшее надежное показание датчика.

Пример

Входные данные	Выходные данные
4 1024 456 1048 234	1024

Исходный код

```
#include <iostream>

using namespace std;

int main()
{
   int n;
   cin>>n;
   int ans=0;
   for(int t=0;t<n;t++)
   {
       int val;
       cin>>val;
       if(val%(16*16)==0)
        ans=max(ans,val);
   }
   cout<<ans;
    return 0;
}
```

Задача №5

Условие

Радиолюбители Петя и Вася увлеклись криптографией и пытаются составить свой неравномерный код для обмена сообщениями. Они оба знают, что для того, чтобы код можно было однозначно прочитать с начала, должно выполняться условие Фано: никакое кодовое слово не должно быть началом предыдущего. Петя переслал Васе кодовую таблицу. Поскольку Петя не очень хорошо работает с ключом, он регулярно сбивался. Когда Петя сбивался, он начинал отправлять кодовое слово заново.

У Васи есть распечатка переданных Петей кодовых слов, содержащая неудачные попытки Пети. Петя составил таблицу правильно, и все кодовые слова удовлетворяют условию Фано. Известно, что Петя очень старается, и поэтому, когда он пытался несколько раз отправить слово, каждая следующая попытка была полнее предыдущей.

Напишите программу, которая найдет, какое наибольшее количество попыток потребовалось Пете, чтобы отправить какое-то кодовое слово

Формат ввода

В первой строке вводится сначала целое число n – количество слов 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 – наибольшее количество попыток Пети отправить кодовое слово.

Примеры

Входные данные	Выходные данные
3 a aab abad	2

4 abc abcd abcde abcfde	3

Исходный код

```
#include <iostream>

using namespace std;

int main()
{
    int n;
    cin>>n;
    char y;
    for(int i=0;i<n;i++)
    cin>>y;
    cout<<n-1;
    return 0;
}
```

Задача №6

Условие

Робот находится посреди плоского зала, расчерченного на квадраты. Будем считать, что стартовая клетка робота имеет координаты (0;0), ось X направлена по направлению «взгляда» робота в начальный момент времени, а ось У – направо от «взгляда» робота в начальный момент времени.

В робота заложена программа, составленная из четырех типов команд.

Вперед: двигает робота вперед на одну клетку вперед

Назад: двигает робота вперед на одну клетку назад

Ось X: поворачивает робота так, что он «смотрит» в направлении оси X

Ось У: поворачивает робота так, что он «смотрит» в направлении оси У

Выяснить, на сколько клеток робот сдвинется по каждой из осей после выполнения программы.

На вход программе подается строка из цифр 1,2,3,4, где 1 означает команду «вперед», 2 – команду «назад», 3 – «ось X», 4 – «ось У». Программа должна вывести два целых числа – смещение по оси X и по оси У.

Пример

Ввод	Вывод
11411	2 2

Исходный код

```
#include <iostream>

using namespace std;

int main()
{
    string s;
    cin>>s;
    int ans[2];
    ans[0]=ans[1]=0;
    int tmp=0;
    for(int i=0;i<s.size();i++)
    {
        if(s[i]=='3')
            tmp=0;
```

```
        if(s[i]=='4')
            tmp=1;
        if(s[i]=='1')
            ans[tmp]++;
        if(s[i]=='2')
            ans[tmp]--;
    }
    cout<< ans[0]<<" "<<ans[1];
    return 0;
}
```

Задача №7

Условие

В браузерной игре, посвященной единоборствам, герой каждый ход может нанести удар по одной из четырех частей тела: голове, груди, животу и ногам. Во время боя случайные события могут помешать ударить какую-то одну часть тела.

Сколько существует последовательностей ударов длиной n, если известна цепочка случайных событий, произошедших за эти ходы?

На вход программе подается строка длиной n (N≤20) символов, состоящая из цифр 0,1,2,3,4.

Символ 0 означает, что случайных событий на этом ходу нет.

Символы с 1 по 4 означают, что нельзя ударить по голове, груди, животу и ногам соответственно.

Вывести целое число – количество последовательностей ударов.

Пример

Ввод	Вывод
11	9

Исходный код

```
#include <iostream>

using namespace std;

int main()
{
    string s;
    cin>>s;
    long long ans=1;
    for(int i=0;i<s.size();i++)
    {
        if(s[i]>='1')
            ans*=3;
        if(s[i]=='0')
            ans*=4;
    }
        cout<<ans;
    return 0;
}
```