Problema - Tribonacci

LOS DATOS SE DEBEN LEER DEL ARCHIVO tribonacci.in LOS RESULTADOS DEBEN GUARDARSE EN UN ARCHIVO LLAMADO respuesta-tribonacci.out

Los números Tribonacci son una generalización de los números de Fibonacci, esta sucesión inicia con $T_1 = 1$, $T_2 = 1$, $T_3 = 2$ y los los siguientes términos son obtenidos sumando de tres en tres.

$$T_4 = T_3 + T_2 + T_1 = 4$$
$$T_5 = T_4 + T_3 + T_2 = 7$$
$$T_6 = T_5 + T_4 + T_3 = 13$$
$$T_7 = T_6 + T_5 + T_4 = 24$$
$$T_8 = T_7 + T_6 + T_5 = 44$$
$$\dots$$

Por ejemplo el Tribonacci 7 es 24.

Dado un número n imprimir el valor de la sucesión en la posición T_n.

Entrada

La entrada consiste de varios casos de prueba. Cada caso de prueba es un numero entero.

Salida

Por cada caso de prueba, en la salida escriba el numero entero que representa en la sucesión de Tribonacci.

Ejemplos de entrada	**Ejemplos de salida**
3 1 3 7	1 2 24

Problema

Para el dato de entrada siguiente, escriba un programa que halle la respuesta.

Programa que resuelve el Problema

```
1
2 #include<iostream>
3 using namespace std;
4
```

```
5  int main() {
6
7    int a;
8    int b;
9    int c;
10   int casos;
11   int d;
12   int i;
13   int n;
14   cin>>casos;
15   for (int j=0;j<casos;j++) {
16     cin>>n;
17     a=1;
18     b=1;
19     c=2;
20     if (n==1) {
21       cout<<a<<endl;
22     } else {
23       if (n==2) {
24         cout<<b<<endl;
25       } else {
26         if (n==3) {
27           cout<<c<<endl;
28         } else {
29           d=0;
30           for (i=4;i<=n;i++) {
31             d=a+b+c;
32             a=b;
33             b=c;
34             c=d;
35           }
36           cout<<d<<endl;
37         }
38       }
39     }
40   }
41
42   return 0;
43 }
```

Problema - Números

LOS DATOS SE DEBEN LEER DEL ARCHIVO numeros.in LOS RESULTADOS DEBEN GUARDARSE EN UN ARCHIVO LLAMADO respuesta-numeros.out

Jorge es un estudiante que fácilmente se aburre en las clases y para no dormirse juega con números, uno de sus juegos favoritos es crear nuevos números a partir de uno por ejemplo si tiene el numero 94857 los nuevos números serán 987 y 45 el primero formado por los dígitos en las las posiciones impares y el segundo formado por los dígitos en las posiciones pares.

Tu tarea es realizar un programa que dado un numero cree dos nuevos utilizando las reglas descritas.

Entrada

La entrada consiste de varios casos de prueba. La primera linea contien el numero de casos de prueba. Para cada caso de prueba hay una linea que contiene un numero entero n ($10 \leq n \leq 99999999$).

Salida

Imprimir los dos nuevos números, cada uno en una linea.

Ejemplos de entrada	**Ejemplos de salida**
3 987 298374 1	97 8 287 934 1 0

Programa que resuelve el Problema

```
1  #include<iostream>
2  using namespace std;
3
4  int main() {
5      int casos;
6      int i,j;
7      string s;
8      cin>>casos;
9      for (j=0;j<casos;j++) {
10         cin>>s;
```

```
11      for (int i = 0; i < s.length(); i+=2)
12          cout<<s.at(i);
13      cout << endl;
14      for (int i = 1; i < s.length(); i+=2)
15          cout<<s.at(i);
16      cout << endl;
17      }
18      return 0;
19
20  }
```

Problema - Abundantes

LOS DATOS SE DEBEN LEER DEL ARCHIVO abundantes.in LOS RESULTADOS DEBEN GUARDARSE EN UN ARCHIVO LLAMADO respuesta-abundantes.out

NumeroLandia, es un lugar donde muchos números pueden convivir, en esta particular historia nos concentraremos en números con abundancia y con deficiencia económica.

Desde el gobierno del país se ha decidido llevar una consulta por zonas de muestra, la intención es decidir dada una zona de muestra cual es la abundancia más grande de los números que viven en la zona. Una zona esta determinada por un par de números que son enteros positivos, e indica que los números comprendidos entre ellos dos conforman la zona de muestra.

Para identificar cual es la mayor abundancia de la zona se debe encontrar los números *abundante*s. Se dice que un número es abundante si la suma de sus divisores exactos, excepto 'el mismo; es mayor que el número. Por ejemplo el número 12 es abundante ya que la suma de sus divisores exactos $1 + 2 + 3 + 4 + 6 = 16$ es mayor a 12, y la abundancia que tiene 12 es 4, que es la diferencia de su *riqueza* respecto a su valor. De los números con deficiencia económica se puede decir que su abundancia es 0.

A partir de esta información se debe encontrar la mayor abundancia que existe en la zona.

Se te pide decidir cuál es la mayor abundancia de una zona de muestra.

Entrada

La entrada consiste de un número entero **n** que indica la cantidad de casos, le siguen **n** líneas con dos números **a, b** que identifican la zona de muestra. $1 \leq a, b \leq 500$.

Salida

La salida es un número entero, que indica la mayor abundancia que existe en la zona de muestra.

Ejemplos de entrada	**Ejemplos de salida**
2 7 18 296 278	4 243

Programa que resuelve el Problema

```
1 #include<iostream>
2 using namespace std;
3
4 int main() {
5
6   int a;
7   int aux;
8   int b;
9   int div;
```

```
10    int i;
11    int j;
12    int mayor;
13    int n;
14    int res;
15    int suma;
16
17    cin>>n;
18    for (i=1;i<=n;i++) {
19      cin>>a>>b;
20      if (a>b) {
21        aux=a;
22        a=b;
23        b=aux;
24      }
25      mayor=0;
26      for (j=a;j<=b;j++) {
27        suma=0;
28        for (div=1;div<=j/2;div++) {
29          if ((j%div)==0) {
30            suma=suma+div;
31          }
32        }
33        if (suma>j) {
34          suma=suma-j;
35        } else {
36          suma=0;
37        }
38        if (suma>mayor) {
39          mayor=suma;
40        }
41      }
42      cout<<mayor<<endl;
43    }
44
45    return 0;
46
47  }
```

Problema - Curioso muy curioso

LOS DATOS SE DEBEN LEER DEL ARCHIVO curioso.in LOS RESULTADOS DEBEN GUARDARSE EN UN ARCHIVO LLAMADO respuesta-curioso.out

En **NumeroLandia**, no podían faltar los números curiosos. Se dice que un número es curioso si la suma del factorial de sus dígitos da el mismo número, por ejemplo 145 es un número curioso, pero el 77 no es número curioso.

La razón por la cual es 145 un curioso es por que: $1! + 4! + 5! = 145$.

Entrada

La entrada consiste de un número entero n, que indica el número de casos, enseguida se tienen n líneas con números enteros positivos que no excedan 10000.

Salida

Para cada caso se debe reportar Y si es curioso y N si no lo es.

Ejemplos de entrada	**Ejemplos de salida**
2 145 77	Y N

Problema

Para el dato de entrada siguiente, escriba un programa que halle la respuesta.

Programa que resuelve el Problema

```
#include<iostream>
using namespace std;

int main() {

  int aux;
  int dig;
  float i;
  float j;
  int n;
  int num;
  int prod;
  string res;
  float suma;

  cin>>n;
```

```
17    for (i=1;i<=n;i++) {
18      cin>>num;
19      aux=num;
20      suma=0;
21      while (num>0) {
22        prod=1;
23        dig=num%10;
24        for (j=1;j<=dig;j++) {
25          prod=prod*j;
26        }
27        suma=suma+prod;
28        num=num/10;
29      }
30      if (suma==aux) {
31        res="Y";
32      } else {
33        res="N";
34      }
35      cout<<res<<endl;
36    }
37
38    return 0;
39
40  }
```

Problema - Cuantos ceros

LOS DATOS SE DEBEN LEER DEL ARCHIVO ceros.in LOS RESULTADOS DEBEN GUARDARSE EN UN ARCHIVO LLAMADO respuesta-ceros.out

El sistema binario es la forma en que las computadoras representan los números. y es una forma de contar números utilizando dos símbolos que son el 1 y el 0. La siguiente tabla muestra los primeros 7 números y su equivalente en binario:

0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111

En este problema no tendrás que convertir números a binario, te darán el equivalente en binario de un numero decimal.

Dado un numero binario X te piden imprimir el numero de bits en 0. Por ejemplo el numero 8 que tiene como su representación binaria 1000 tiene 3 dígitos en 0. No se consideran los ceros que están a la izquierda del primer número uno.

Entrada

La entrada consiste de multiples casos de prueba. La primera linea contiene el numero N de casos de prueba. Luego siguen varias lineas, cada una es un caso de prueba que consiste de un numero entero X.

Salida

Por cada caso de prueba, imprima en una linea el numero de ceros que tiene la representación binaria del numero X.

Ejemplos de entrada	**Ejemplos de salida**
3 111 001000 1010	0 3 2

Programa que resuelve el Problema

```
#include<iostream>
#include<string>
using namespace std;

int main() {

    int c,sw;
    string x;
    int n;
    cin>>n;
    for (int i=0;i<n;i++){
        cin>>x;
    c=0;
    sw=0;
    for(int j=0;j<x.size();j++){
       if(x.at(j)=='1')
          sw=1;
       if(sw==1)
         if(x.at(j)=='0')
            c=c+1;
        }
    cout<<c<<endl;
    }
    return 0;
}
```

Problema - Emparejando Peones

LOS DATOS SE DEBEN LEER DEL ARCHIVO peones.in LOS RESULTADOS DEBEN GUARDARSE EN UN ARCHIVO LLAMADO respuesta-peones.out

Emparejar peones es un juego que se juega en una cinta de papel dividida en N celdas que están etiquetadas de 0 a $N-1$. Cada celda tiene un numero arbitrario de peones.

El objetivo del juego es traer el mayor numero de peones a la celda 0. Las únicas movidas validas son:

- Encontrar un par de peones que compartan una celda digamos X.
- Quitar un par de peones de la celda X.
- Adicionar un peón a la celda $X-1$.

Usted puede realizar tantas movidas como desee en cualquier orden.

Al final se pide mostrar cuantos peones hay en la celda 0.

Entrada

La entrada consiste de múltiples casos de prueba. La primera linea contien el numero de casos de prueba. Cada caso de prueba tiene dos lineas.

La primera linea contiene el numero de celdas N. La segunda linea contiene N números que representan el numero de peones que hay en cada celda.

Salida

Muestre en una linea cuantos peones hay en la celda 0.

Ejemplos de entrada

```
6
2
0 2
2
10 3
4
0 0 0 8
4
0 1 1 2
19
```

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 123456
8
1000 2000 3000 4000 5000 6000 7000 8000
```

Ejemplos de salida

```
1
11
1
1
0
3921
```

Programa que resuelve el Problema

```
#include<iostream>
using namespace std;

int main() {
  int casos;
  int i,j,n,v;
  cin>>casos;
  for (j=0;j<casos;j++) {
    cin>>n;
        long long s[n];
    for (i=0;i<n;i++) {
      cin>>s[i];
    }
    for (i=n-1;i>0;i--) {
      if (s[i]>1) {
        s[i-1]=s[i-1]+s[i]/2;
      }
    }
    cout<<s[0]<<endl;
  }

  return 0;

}
```

Problema - Palabras Palindrome

LOS DATOS SE DEBEN LEER DEL ARCHIVO palindrome.in LOS RESULTADOS DEBEN GUARDARSE EN UN ARCHIVO LLAMADO respuesta-palindrome.out

Una palabra se dice que es palindrome si se puede leer de la misma forma de izquierda a derecha y de derecha a izquierda. Por ejemplo las cadenas $ANA, AAXAA, Z, XYYYYYX$ son palindromes.

Llamaremos palabras palindromes faux si después de remplazar un grupo de letras consecutivas e iguales con una sola letras, es un palindrome.

Por ejemplo la cadena $AAAAANNAA$ no es un palindrome. Ahora si remplazamos un grupo de letras consecutivas $AAAAA$ por A y NN por N y AA por A obtenemos ANA que es un palindrome.

Entrada

La entrada consiste de varios casos de prueba. La primera linea contiene el numero casos de prueba. Cada caso de prueba esta en una línea y consiste de una cadena que tiene un máximo de 50 caracteres. Las letras pueden ser de la A a la Z.

Salida

Por cada linea de entrada imprima una linea de salida con las palabras

- $PALINDROME$, si la palabra es un palindrome.
- $FAUX$, si la palabra no es palindrome pero es un palindrome faux.
- $NOTEVENFAUX$ en caso de que la palabra no pes palindrome o palindrome faux.

Ejemplos de entrada	**Ejemplos de salida**
7 ANA AAAAANNAA LEGENDARY XXXYTYYTTYX TOPCOODEREDOOCPOT TOPCODEREDOOCPOT XXXXYYYYYZZXXYYY	PALINDROME FAUX NOT EVEN FAUX FAUX PALINDROME FAUX NOT EVEN FAUX

Programa que resuelve el Problema

```
#include<string>
#include<iostream>
using namespace std;

bool palindrome(string s){
    bool t=true;
    int l=s.size();
    int j=0;
    int c1,c2;
    for (int i=0;i<l/2;i++) {
        j=l-i-1;
        c1=s.at(i);
        c2=s.at(j);
        if (c1 != c2) {
            t=false;
        }
    }
    return t;
}
int main() {
    int casos;
    cin>>casos;
    for(int j1=0;j1<casos;j1++){
        string s;
        string x;
        cin>>s;
    int i,j,l;
    char ant;
    if (palindrome(s)) {
        cout<<"PALINDROME"<<endl;
    } else {
        x=s.at(0);
        ant=s.at(0);
        for (i=0;i<s.size();i++) {
            if(s.at(i)!=ant){
                x+=s.at(i);
                ant=s.at(i);
            }
        }
        if (palindrome(x)) {
            cout<<"FAUX"<<endl;
        } else {
            cout<<"NOT EVEN FAUX"<<endl;
        }
    }
```

```
46      }
47
48      return 0;
49  }
```

Problema - Adivinando el Examen

LOS DATOS SE DEBEN LEER DEL ARCHIVO examen.in LOS RESULTADOS DEBEN GUARDARSE EN UN ARCHIVO LLAMADO respuesta-examen.out

El señor tramposo esta tomando parte en un examen en linea. El examen consiste en N preguntas de respuesta multiple convenientemente numeradas de 0 a $N-1$. Para cada pregunta las respuestas están etiquetadas como A, B, C. Para cada pregunta una de las tres posibles respuesta es correcta. A usted le dan las información correcta de las respuestas, pero, el señor tramposo no sabe.

El señor tramposo en su desesperación ha hack-eado el sitio web del examen. Desafortunadamente lo unico que pudo obtener es información agregada sobre los respuestas que se utilizarán. Más precisamente conoce cuantas respuestas son A, cuantas B y cuantas C.

Cuando toma el exámen recibe una pregunta y por cada pregunta el utiliza su intuicion para escoger una respuesta. Una vez que contesta una pregunta se le informa inmediatamente la respuesta correcta.

Algunas veces el el puede encontrar una regla para encontrar la respuesta correcta. Lo que se quiere determinar cuantas opciones tiene el señor tramposo para contestar cara una de las preguntas.

Por ejemplo, si tenemos las respuestas posible $AAABBB$, cococe que hay tres A y tres B. Para la primera pregunta el puede contestar A o B. Luego se le informa que la respuesta es A. Para esta pregunta que tiene dos opciones. Cuando le presentan la segunda pregunta, el sabe que salio una letra A pero aún tiene dos opciones para esta pregunta. Cuando sale la cuarta pregunta el sabe que no hay más A disponibles y solo le queda una posibilidad la B. El resultado sera $2, 2, 2, 1, 1, 1$

Entrada

La entrada consiste en multiples casos de prueba. La primera linea indica cuantos casos de prueba hay. Cada caso de prueba viene en una linea, que es una cadena con las respuestas a las preguntas.

Salida

En la salida escriba un numero separado por espación que indica el numero de posibles respuesta que hay para una pregunta dada.

Ejemplos de entrada

```
5
AAAAA
```

```
AAABBB
CAAAAAC
BBCA
BACACABCBBBBCAAAAACCCABBCAA
```

Ejemplos de salida

```
[1, 1, 1, 1, 1]
[2, 2, 2, 1, 1, 1]
[2, 2, 2, 2, 2, 2, 1]
[3, 3, 2, 1]
[3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 2, 1, 1]
```

Programa que resuelve el Problema

```
#include<iostream>
using namespace std;

int main() {
    int casos;
    int i,j,v;
    string s;
    cin>>casos;
    for (j=0;j<casos;j++) {
        cin>>s;
    int r[3];
    int resp[s.size()];
    r[0]=0;
    r[1]=0;
    r[2]=0;
    for (int i = 0; i < s.length(); i++)
        resp[i]=0;
    for (int i = 0; i < s.length(); i++) {
        r[s.at(i)-'A']++;
    }
    for (int i = 0; i < s.length(); i++) {
        if (r[0]>0)
            resp[i]++;
        if (r[1]>0)
            resp[i]++;
        if (r[2]>0)
            resp[i]++;
```

```
28          r[s.at(i)-'A']--;
29      }
30
31      cout<<"[";
32      for (int i = 0; i < s.length(); i++) {
33          cout<<resp[i];
34          if (i < s.length()-1)
35              cout<<", ";
36      }
37      cout<<"]"<<endl;
38      }
39      return 0;
40
41  }
```

Problema - Juego de Vida

LOS DATOS SE DEBEN LEER DEL ARCHIVO vivir.in LOS RESULTADOS DEBEN GUARDARSE EN UN ARCHIVO LLAMADO respuesta-vivir.out

Un conejo y un gato inventaron una variación al Juego de la Vida.

Se tienen N celdas acomodadas en un circulo. Estas celdas están numeradas del 0 al $N-1$. Para todas las celdas entre i y $N-2$, inclusive, la *iesima* celda es adyacente a la celda $i-1$ la celda $N-1$ y la celda 0 son adyacentes. Cada celda tiene exactamente dos celdas adyacentes. Cada celda puede tener dos estados *vivir*, o *morir*.

El gato y el conejo pueden decidir el estado de las celdas en el instante $t=0$. Para $t>0$ los estados se determinan como sigue:

- Considere tres celdas: la celda *iesima* y sus dos celdas adyacentes.
- Si al menos dos de los tres celtas son *vivir* en el tiempo $t-1$ el estado de la celda *iesima* será *vivir*
- Si al menos dos de la tres celdas son *morir* en el tiempo $t-1$ el estado de la celda *iesima* será *morir*

En el tiempo inicial se tiene una cadena donde cada caracter representa una celda. La *iesima* celda esta representada por el caracter 0, que significa *morir* o el caracter 1 que significa *vivir*.

Dado el tiempo t se le pide hallar el estado el estado final.

Por ejemplo: dada la cadena 01010 en el tiempo $t=1$ la cadena se vuelve 00100, en el tiempo $t=2$ se convierte en 00000. Esta es la respuesta para $t=2$.

Entrada

La entrada consiste en varios casos de prueba y termina cuando no hay más datos. Cada caso de prueba esta en dos lineas la primera linea que contiene la cadena don los estados iniciales y la segundo el valor de t.

Salida

Escriba en una linea en el mismo formato de la entrada el estado al que se llegara en el tiempo t.

Ejemplos de entrada	**Ejemplos de salida**
6 01010 2 010101 5 111010 58 111111111 511 11001000001011111001010000 1001 1000 00101110011 0	00000 101010 111111 111111111 11000000000111111000000000 0001 00101110011

Programa que resuelve el Problema

```
1  #include<iostream>
2  #include<string>
3  using namespace std;
4
5  int main() {
6      char vivir='1';
7      char morir='0';
8      int n,t;
9      string s;
10     cin>>n;
11     for (int i=0;i<n;i++){
12         cin >> s;
13     cin >> t;
14     int l=s.size();
15     char estado [l];
16     for (int i = 0; i < l; i++)
17         estado[i]=s.at(i);
18     int suma=0;
19     for (int j=0 ; j<t; j++){
20         for (int i = 0; i < l; i++) {
21             suma=0;
22             if (i==0)
23                 suma=estado[l-1]+estado[i]+estado[i+1]-(3*morir);
24             else
25                 if(i==l-1)
26                     suma=estado[i-1]+estado[i]+estado[0]-(3*morir);
27                 else
28                     suma=estado[i-1]+estado[i]+estado[i+1]-(3*morir)
                           ;
```

```
29              if (suma>1)
30                  estado[i]=vivir;
31              else
32                  estado[i]=morir;
33          }
34      }
35      for (int i = 0; i < l; i++)
36          cout << estado[i];
37      cout << endl;
38      }
39      return 0;
40
41  }
```

Proponentes de los Problemas

- Tribonacci - Hernan Payrumani
- Números - Hernan Payrumani
- Abundantes - Leticia Blanco
- Curioso - Leticia Blanco
- Ceros - Jorge Teran
- Peones - Jorge Teran
- Palindrome - Jorge Teran
- Examen - Jorge Teran
- Vivir - Jorge Teran