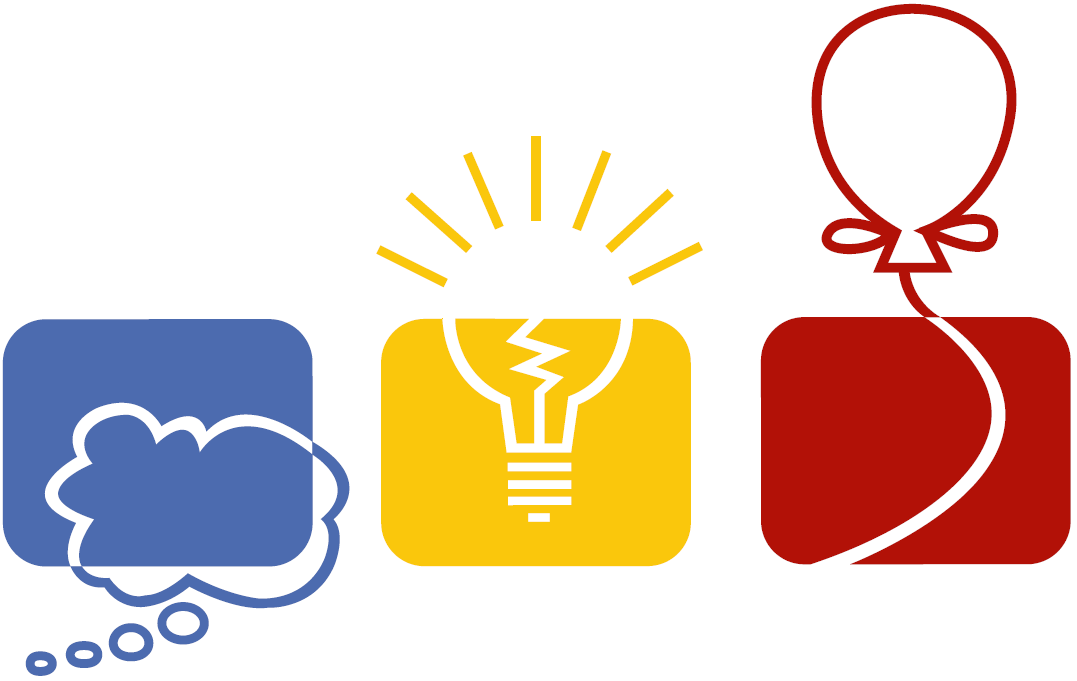
**UNIVERSIDAD PRIVADA DE TACNA**

**FACULTAD DE INGENIERÍA**

** **

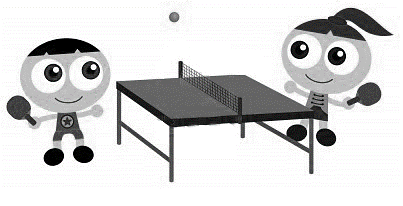
**PRIMER CONCURSO DE PROGRAMACIÓN**

**PROBLEM SET**

1. **Asignando Equipos**
2. **Polyhedra**
3. **Mayoría**
4. **Lenguas**
5. **Balas de Cañón**

8 de junio de 2018

**Asignando Equipos**



Cuatro amigos juegan al tenis de mesa. Cada uno de ellos tiene un nivel de habilidad que está representado por un número entero: cuanto mayor sea el número, mejor será el jugador.

Los cuatro amigos quieren formar dos equipos de dos jugadores cada uno. Para que el juego sea más emocionante quieren que el nivel de habilidad de los equipos sea lo más cercano posible. El nivel de habilidad de un equipo es la suma de los niveles de habilidad de los jugadores en ese equipo.

A pesar de que son muy buenos jugadores de tenis de mesa, estos amigos no son tan buenos en otras cosas, como Matemáticas o Computación. ¿Puedes ayudarlos a encontrar la menor diferencia posible entre las habilidades de los equipos?

**Input**

La entrada consta de una sola línea que contiene cuatro enteros A, B, C y D, que representan la habilidad de los cuatro jugadores (0 ≤ A ≤ B ≤ C ≤ D ≤ 10004).

**Output**

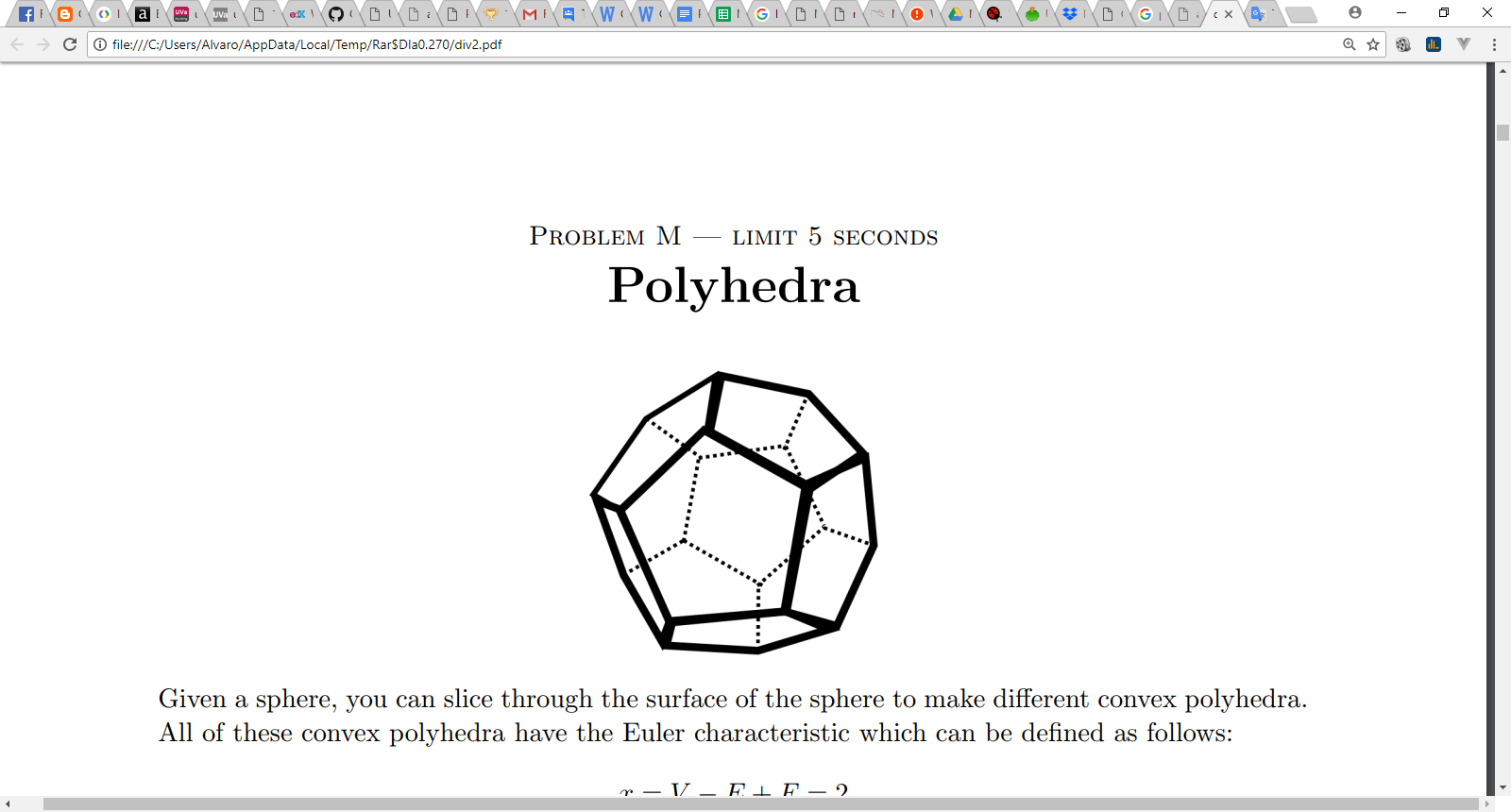
Imprime una línea con un número entero que representa la diferencia más pequeña entre los niveles de habilidad para ambos equipos.

|  |  |
| --- | --- |
| **Sample Input 1**  4 7 10 20 | **Sample Output 1**  7 |

|  |  |
| --- | --- |
| **Sample Input 2**  0 0 1 1000 | **Sample Output 2**  999 |

|  |  |
| --- | --- |
| **Sample Input 3**  1 2 3 4 | **Sample Output 3**  0 |

**Polyhedra**



Dada una esfera, puedes cortar a través de la superficie de la esfera para crear diferentes poliedros convexos.

Todos estos poliedros convexos tienen la característica de Euler que se puede definir de la siguiente manera:

x = V − E + F = 2

donde V representa el número de vértices, E el número de bordes y F el número de caras en un poliedro convexo.

**Input**

La entrada comienza con una línea con un entero único T, 1 ≤ T ≤ 100, que denota el número de casos de prueba.

Cada caso de prueba consta de una sola línea con dos enteros separados por espacios V y E (4 ≤ V, E ≤ 100),

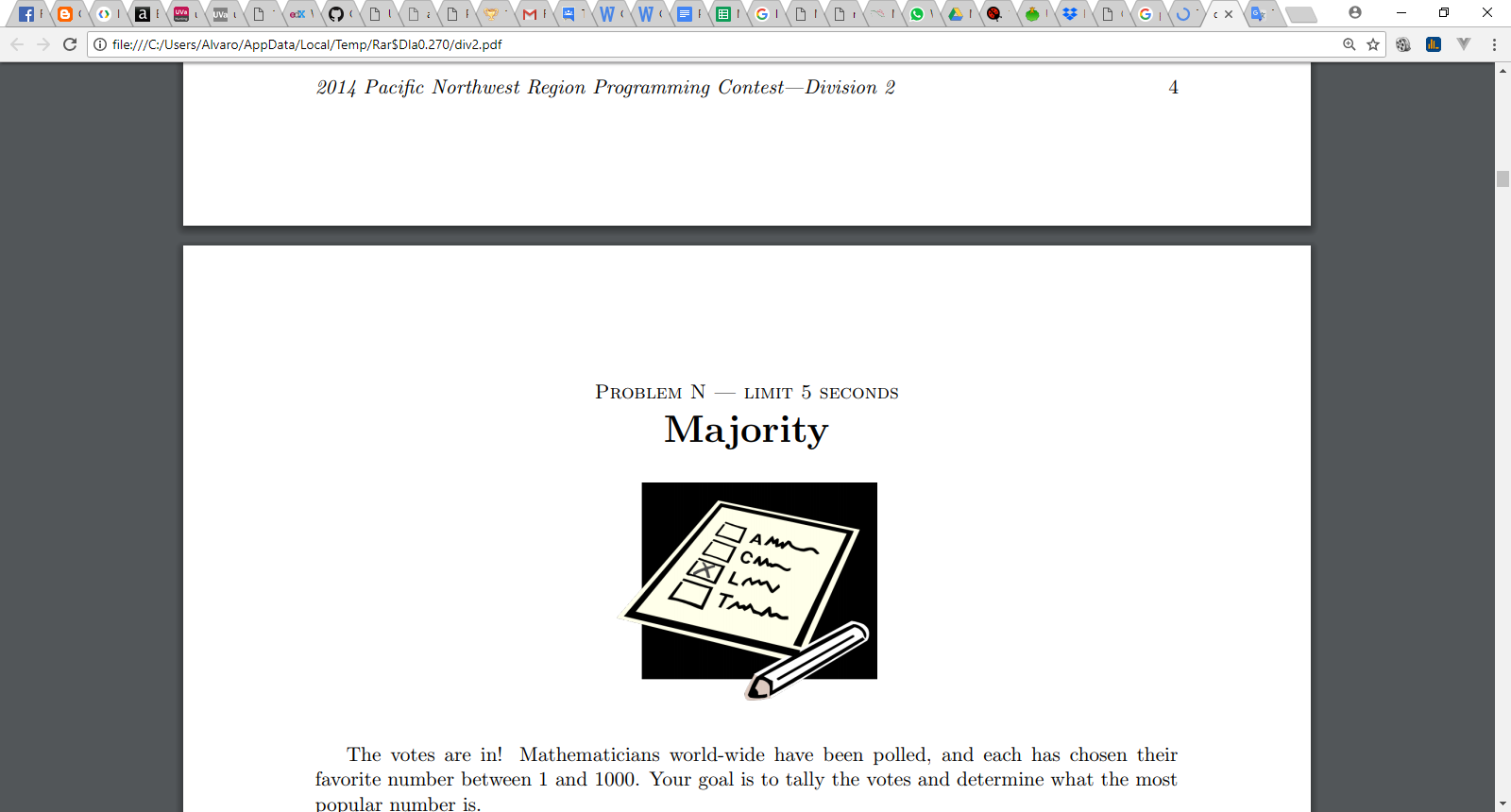
que representa el número de vértices y bordes respectivamente del poliedro convexo.

**Output**

Para cada caso de prueba, imprima en una sola línea el número de caras en el poliedro definido.

|  |  |
| --- | --- |
| **Sample Input**  2  8 12  4 6 | **Sample Output**  6  4 |

**Mayoría**



¡Los votos están listos! Los Docente de FAING son conocidos por que les gusta las matemáticas, han sido encuestados, y cada uno ha elegido su número favorito entre 1 y 1000. Su objetivo es contar los votos y determinar cuál es el número más popular.

Si hay un empate para la mayor cantidad de votos, elija el número más pequeño con tantos votos.

**Input**

La entrada comenzará con una sola línea que contiene el número de casos de prueba, entre 1 y 100 inclusive.

Para cada caso de prueba, habrá una sola línea que dará el número de votos V, 1 ≤ V ≤ 1000. A continuación, V líneas, cada una con un voto entero único entre 1 y 1000.

**Output**

Para cada caso de prueba debe imprimir el número más popular.

|  |  |
| --- | --- |
| **Sample Input**  3  3  42  42  19  4  7  99  99  7  5  11  12  13  14  15 | **Sample Output**  42  7  11 |

**Lenguas**

Los escritos de Gandalf han estado disponibles durante mucho tiempo para el estudio, pero nadie ha descubierto aún qué idioma en el que están escritos. Recientemente, debido al trabajo de programación de un hacker conocido solo por

el nombre en clave ROT13, se descubrió que Gandalf no usó más que una simple letra

esquema de sustitución, además que es su propia inversa: la misma operación codifica el mensaje como lo descifra.

Esta operación se realiza reemplazando vocales en la secuencia

(a i y e o u)

con la vocal tres veces avanzada cíclicamente.

De manera similar, las consonantes son reemplazadas de la secuencia

(b k x z n h d c w g p v j q t s r l m f)

avanzando diez letras. Entonces, por ejemplo, la frase:

One ring to rule them all.

se traduce a:

Ita dotf ni dyca nsaw ecc.

Lo fascinante de esta transformación es que el lenguaje resultante produce palabras pronunciables.

Para este problema, escribirás código para traducir los manuscritos de Gandalf a texto sin formato.

**Input**

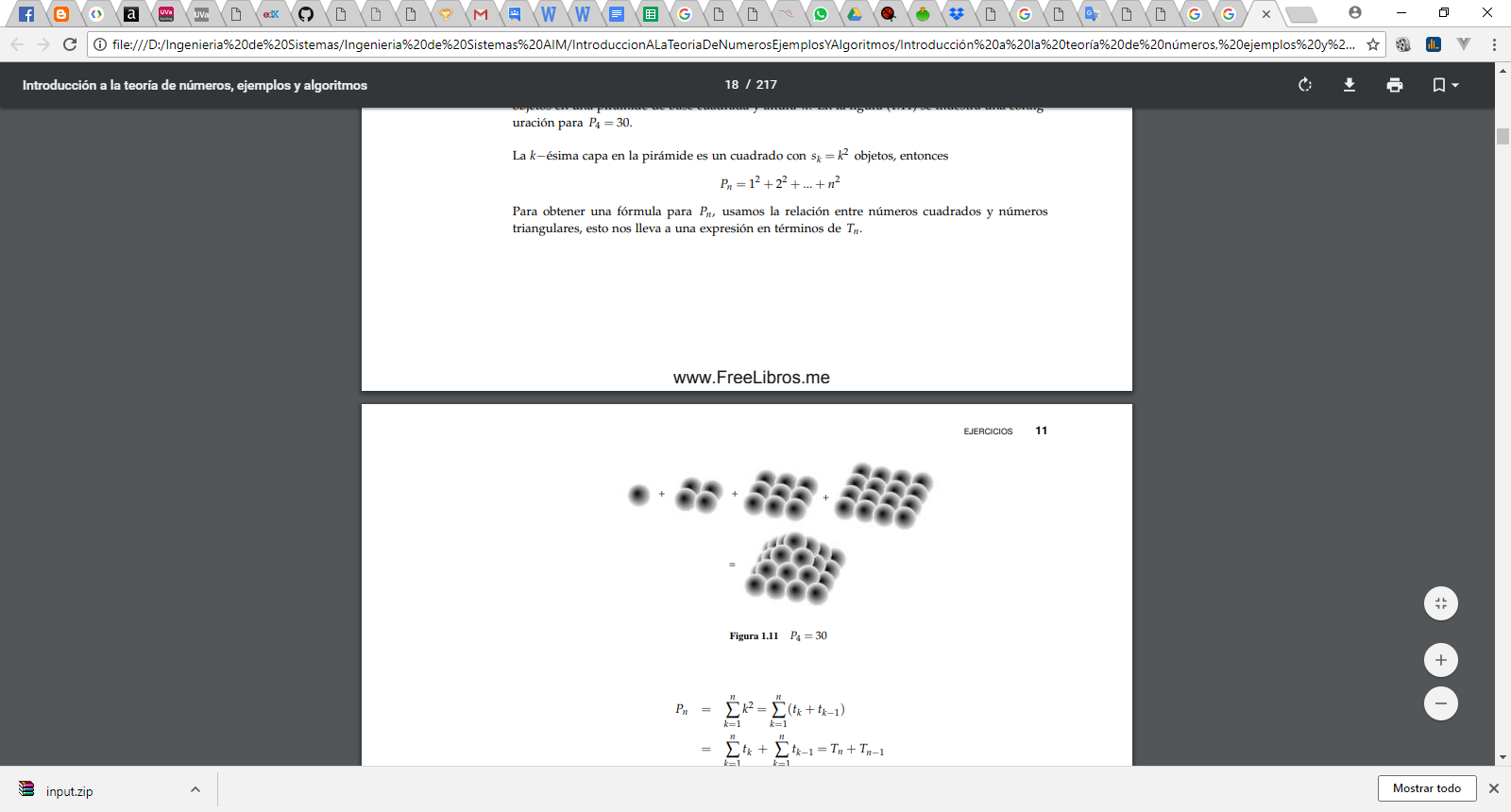
El archivo de entrada contendrá múltiples casos de prueba. Cada caso de prueba consiste en una sola línea que contiene hasta 100 caracteres, que representan algún texto escrito por Gandalf. El final de la entrada se denota por el final del archivo.

**Output**

Para cada caso de prueba de entrada, imprima su traducción en texto claro. La salida debe contener exactamente la misma cantidad de líneas y caracteres que la entrada.

|  |  |
| --- | --- |
| **Sample Input**  Ita dotf ni dyca nsaw ecc. | **Sample Output**  One ring to rule them all. |

**Balas de Cañón**



Pedrito mientras limpia y organiza la Tierra despoblada, se ha topado con algunos monumentos conmemorativos de la batalla del Alto de la Alianza.

Él está consolidando las balas de cañón en un solo lugar, y decide usar las pirámides con bases triangulares en vez de aquellas con bases cuadradas.

En los monumentos las balas de cañón a veces se apilaban como pirámide de cuatro lados, con la base como un cuadrado de balas de cañón con n bolas en cada lado. Una alternativa es apilarlos en una pirámide de tres lados, que es de hecho, uno de los sólidos platónicos, un tetraedro.

Este tetraedro de balas de cañón tiene una base que es un triángulo equilátero de balas de cañón con n bolas en cada lado. El número de bolas en ese triángulo se da simplemente sumando los números de 1 a n. En la parte superior de cada capa (comenzando desde la base) es un triángulo con una bola menos en cada lado, hasta la capa superior con una sola bala.

Dada la cantidad de balas de cañón en cada lado de la base, calcule el número total de balas de cañón en la totalidad de la pila tetraédrica.

**Input**

La primera línea contiene un número único n, que da la cantidad de problemas tetraédricos planteados, para un máximo de 100 problemas, A continuación, hay exactamente n líneas, cada una con un solo número que da el número de balas de cañón a cada lado de la base para un tetraedro de balas de cañón, un número mayor que 0 y menor que 100000.

**Output**

Para cada problema, muestre el número del problema (comenzando desde 1), dos puntos y un espacio en blanco, el número de balas de cañón a cada lado de la base, un espacio en blanco, y finalmente el número total de balas de cañón en el tetraedro.

|  |  |
| --- | --- |
| **Sample input**  6  1  2  3  5  27  999 | **Sample output**  1: 1 1  2: 2 4  3: 3 10  4: 5 35  5: 27 3654  6: 999 166666500 |

**Soluciones**

**Asignando Equipos**

#include<bits/stdc++.h>

using namespace std;

int main(){

int a,b,c,d;

cin>>a>>b>>c>>d;

cout<<abs(a+d-b-c)<<endl;

return 0;

}

**Polyhedra**

#include<bits/stdc++.h>

using namespace std;

int main(){

int n;

cin>>n;

while(n--){

int a,b;

cin>>a>>b;

cout<<b-a+2<<endl;

}

return 0;

}

**Mayoría**

#include<bits/stdc++.h>

using namespace std;

int main(){

int t,n,a;

cin>>t;

while(t--){

cin>>n;

int maxi=-1;

int votos[1001]={0};

while(n--){

cin>>a;

votos[a]++;

if(votos[a]>maxi)maxi=votos[a];

}

for(int i=1;i<1001;i++){

if(votos[i]==maxi){

cout<<i<<endl;

break;

}

}

}

return 0;

}

**Lenguas**

#include<bits/stdc++.h>

using namespace std;

int main(){

string cad;

while(getline(cin,cad)){

for(int i=0;i<cad.size();i++){

switch (cad[i]){

case 'a':

cout<<'e';

break;

case 'A':

cout<<'E';

break;

case 'E':

cout<<'A';

break;

case 'e':

cout<<'a';

break;

case 'i':

cout<<'o';

break;

case 'I':

cout<<'O';

break;

case 'o':

cout<<'i';

break;

case 'O':

cout<<'I';

break;

case 'Y':

cout<<'U';

break;

case 'y':

cout<<'u';

break;

case 'U':

cout<<'Y';

break;

case 'u':

cout<<'y';

break;

case 'f':

cout<<'g';

break;

case 'F':

cout<<'G';

break;

case 'm':

cout<<'w';

break;

case 'M':

cout<<'W';

break;

case 'l':

cout<<'c';

break;

case 'L':

cout<<'C';

break;

case 'r':

cout<<'d';

break;

case 'R':

cout<<'D';

break;

case 's':

cout<<'h';

break;

case 'S':

cout<<'H';

break;

case 't':

cout<<'n';

break;

case 'T':

cout<<'N';

break;

case 'q':

cout<<'z';

break;

case 'Q':

cout<<'Z';

break;

case 'j':

cout<<'x';

break;

case 'J':

cout<<'X';

break;

case 'v':

cout<<'k';

break;

case 'V':

cout<<'K';

break;

case 'p':

cout<<'b';

break;

case 'P':

cout<<'B';

break;

case 'g':

cout<<'f';

break;

case 'G':

cout<<'F';

break;

case 'w':

cout<<'m';

break;

case 'W':

cout<<'M';

break;

case 'c':

cout<<'l';

break;

case 'C':

cout<<'L';

break;

case 'd':

cout<<'r';

break;

case 'D':

cout<<'R';

break;

case 'H':

cout<<'S';

break;

case 'h':

cout<<'s';

break;

case 'n':

cout<<'t';

break;

case 'N':

cout<<'T';

break;

case 'z':

cout<<'q';

break;

case 'Z':

cout<<'Q';

break;

case 'x':

cout<<'j';

break;

case 'X':

cout<<'J';

break;

case 'k':

cout<<'v';

break;

case 'K':

cout<<'V';

break;

case 'B':

cout<<'P';

break;

case 'b':

cout<<'p';

break;

default:

cout<<cad[i];

break;

}

}

cout<<"\n";

}

return 0;

}

**Balas de Cañón**

#include<bits/stdc++.h>

using namespace std;

int main(){

long long t,n;

cin>>t;

for(int k=1;k<=t;k++){

cin>>n;

long long b=0,b1;

for(int i=1;i<=n;i++){

b1=(long long)i\*(long long)(i+1)/2LL;

b+=b1;

}

cout<<k<<": "<<n<<" "<<b<<"\n";

}

return 0;

}