Soluciones a Problemas de: Club de Programación Competitiva UNAM - Primer sesión

Pu++

31 de marzo del 2020

FizzBuzz

El problema nos pide imprimir los números del 1 al 1000. Debemos imprimir "Fizz"cuando el número es múltiplo de 3, "Buzz"cuando es múltiplo de 5 y "FizzBuzz"cuando es múltiplo de 15.

FizzBuzz ii

```
#include <bits/stdc++.h>
using namespace std;

int main(void){
    for(int i=1;i<=1000;i++){
        if(i%15==0){
            cout<<"FizzBuzz"<<endl;
        }else if(i%3==0){
            cout<<"Fizz"<<endl;
        }else if(i%5==0){
            cout<<"Buzz"<<endl;
        }
        else{cout<<i<<endl};
    }
    return 0;
}
```

Intervalos

Intervalos i

En este problema tenemos que leer a, b y c, y determinar si c está en el intervalo [a,b], a su izquierda o a su derecha.

Hay muchas formas de hacer esto. Una muy simple es la siguiente:

- Si $c < a$, entonces está a la izquierda.
- Si $c > b$, entonces está a la derecha.
- En otro caso está en el intervalo.

Intervalos ii

```cpp
#include <bits/stdc++.h>
using namespace std;

int main(void){
    int a, b, c;
    cin>>a>>b>>c;
    if(c < a){
        cout<<"IZQUIERDA"<<endl;
    }else if(c > b){
        cout<<"DERECHA"<<endl;
    }else{
        cout<<"INTERVALO"<<endl;
    }
    return 0;
}
```

Tabla para Pastel

Tabla para Pastel i

Queremos calcular el área del cuadro azul.

Tabla para Pastel ii

Sabemos que **n** es el área del cuadro blanco. Tenemos que encontrar una relación entre ambos cuadros para lograr sacar el área azul. Llamémosle a un lado del cuadro azul **d**, notemos que el radio del pastel **r**, es la mitad de **d**. Ya que el diámetro del pastel es igual a los lados del cuadrado azul. Entonces, necesitamos $d * d$. Si tenemos **r**, tenemos el área deseada: $2r * 2r = 4r^2$.

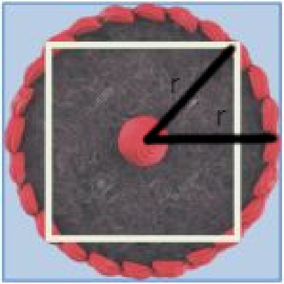
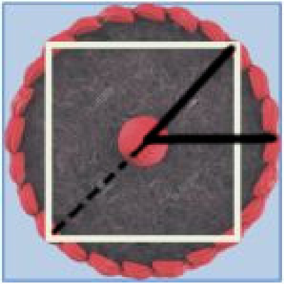


Tabla para Pastel iii

De la secundaria recordarás el Teorema de Pitágoras: $a^2 + b^2 = c^2$. Lo ocuparemos porque hay que obtener **r**. Hay que notar que el radio del pastel es la mitad de la hipotenusa del triangulo rectángulo formado por el cuadro blanco de lados $\sqrt{n}$.



Muy bien, si sacamos la hipotenusa tenemos el diámetro, entonces tenemos el radio y tenemos la respuesta deseada. Hay que ponerlo en una fórmula:

$$a^2 + b^2 = c^2$$

$$(\sqrt{n})^2 + (\sqrt{n})^2 = (d)^2$$

$$(\sqrt{n})^2 + (\sqrt{n})^2 = (2r)^2$$

$$2(\sqrt{n})^2 = (2r)^2$$

$$2n = (2r)^2$$

$$2n = 4r^2$$

Tabla para Pastel v

```cpp
#include <bits/stdc++.h>
using namespace std;

int main() {
    int T;
    cin >> T;
    while (T > 0) {
        long long n;
        cin >> n;
        cout << "Area a comprar: " << n * 2 << endl;
        T = T - 1;
    }
}
```

La visión de Ana

La visión de Ana i

En este problema tenemos que usar *doubles*. El problema se traduce a encontrar el día en el que el precio del dólar era el mínimo.

La visión de Ana ii

```
#include <bits/stdc++.h>
using namespace std;

int main(void){
    int t;
    cin>>t;
    for(int cases=0;cases<t;cases++){
        int s, n;
        cin>>s>>n;
        double minimo = 101.0; //hacemos el minimo un número grande
        int dia=-1;
        for(int i=0;i<n;i++){
            double costo;
            cin>>costo;
            if(costo < minimo){
                minimo = costo;
```

La visión de Ana iii

```
                dia = i+1;
            }
        }
        cout<<"Case "<<cases+1<<": comprar en dia "<<dia<<endl;
    }
    return 0;
}
```

Barbulla Matemática

Queremos calcular:

$$S_n = 1 + (1+2) + (1+2+3) + (1+2+3+4) + \ldots + (1+2+\ldots+n)$$

Tenemos la fortuna de saber sobre la suma de Gauss:

$$(1 + 2 + \ldots + n) = \frac{n(n+1)}{2}$$

Una solución sencilla sería hacer n sumas de Gauss desde 1 hasta n. Podemos lograr eso con un ciclo y acumulando la suma de las sumas de Gauss en una variable.

Barbulla Matemática ii

```
#include <bits/stdc++.h>
using namespace std;

int main(){
    int n, total;
    cin >> n;
    total = 0;
    for (int i = 0; i <= n; i++){
        total += i*(i+1)/2;
    }
    cout << total << endl;
    return 0;
}
```

Nota: `a += b;` es una forma corta de escribir `a = a + b;`

La solución dada resuleve el problema, sin embargo a continuación se presenta como **bonus** una solución más eficiente para cualquier interesado.

Barbulla Matemática iv

Intentemos darle un giro más matemático al asunto.

Primero hablemos sobre las combinaciones.
El número de k-combinaciones en un conjunto de n elementos es igual al coeficiente binomial $\binom{n}{k}$. Es decir:

$\binom{n}{k} = \frac{n!}{k!(n-k)!}$ — number of ways to choose k objects out of n

Entonces podemos ver que: $\binom{n+1}{2} = \frac{(n+1)!}{(n-1)!2!} = \frac{(n+1)(n)(n-1)!}{(n-1)!2!} = \frac{n(n+1)}{2}$

Así que nuestro problema se traduce en resolver:
$\binom{2}{2} + \binom{3}{2} + \binom{4}{2} + \binom{5}{2} + \ldots + \binom{n+1}{2}$

Barbulla Matemática v

Ahora, necesitamos saber 2 cosas.

La primera es que: $\binom{a}{a} = 1 = \binom{b}{b}$
Ya que solo hay una forma de tomar todos los elementos de un conjunto.

Con ésto podemos traducir el problema a:

$$\underline{\binom{2}{2}} + \binom{3}{2} + \binom{4}{2} + \binom{5}{2} + \ldots + \binom{n+1}{2} = \underline{\binom{3}{3}} + \binom{3}{2} + \binom{4}{2} + \binom{5}{2} + \ldots + \binom{n+1}{2}$$

Barbulla Matemática vi

La segunda es que:

$$\text{Fórmula de Pascal: } \binom{n+1}{k+1} = \binom{n}{k} + \binom{n}{k+1}$$

Se puede ver desarrollando algebraicamente o con combinatoria enumerativa.

Si usamos lo anterior, podemos ver que:

$$\underline{\binom{3}{2} + \binom{3}{3}} + \binom{4}{2} + \binom{5}{2} + \ldots + \binom{n+1}{2} = \underline{\binom{4}{3}} + \binom{4}{2} + \binom{5}{2} + \ldots + \binom{n+1}{2}$$

$$\underline{\binom{4}{2} + \binom{4}{3}} + \binom{5}{2} + \ldots + \binom{n+1}{2} = \underline{\binom{5}{3}} + \binom{5}{2} + \ldots + \binom{n+1}{2}$$

$$\underline{\binom{5}{2} + \binom{5}{3}} + \ldots + \binom{n+1}{2} = \ldots = \underline{\binom{n+1}{3} + \binom{n+1}{2}} = \underline{\binom{n+2}{3}}$$

Y sabemos que : $\binom{n+2}{3} = \frac{(n+2)!}{(n-1)!3!} = \frac{(n+2)(n+1)(n)(n-1)!}{(n-1)!3!} = \frac{(n+2)(n+1)n}{6}$

Solución final:

```
#include <bits/stdc++.h>
using namespace std;

int main(){
  int n;
  cin >> n;
  cout << n*(n+1)*(n+2)/6 << endl;
  return 0;
}
```

On ta el pinche fácil pa irme?

On ta el pinche fácil pa irme? i

En este problema, lo único que tenemos que calcular es n*m.

¡Cuidado! Si no nos fijamos en los tamaños de n y m, la solución será sólo parcialmente correcta.

Como n y m pueden ser muy grandes, su multiplicación puede que sea más grande que la capacidad de un int.

Tenemos entonces que utilizar long long.

On ta el pinche fácil pa irme? ii

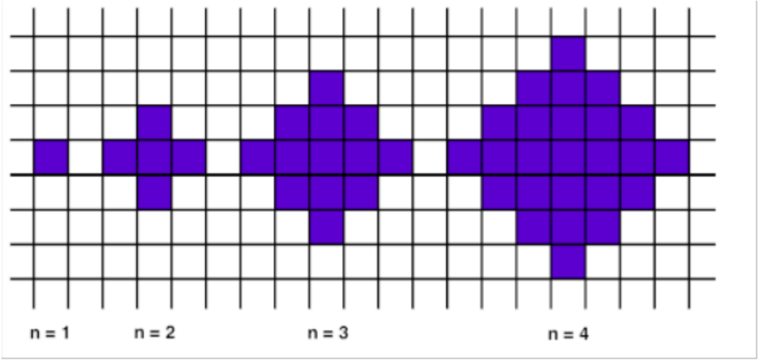
```cpp
#include <bits/stdc++.h>
using namespace std;

int main(void){
    long long n, m;
    cin>>n>>m;
    cout<<n*m<<endl;
    return 0;
}
```

Área de la figura

Área de la figura i

Queremos calcular el área de las figuras:



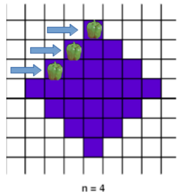
Área de la figura ii

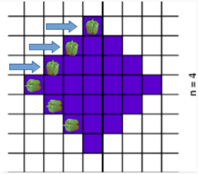
Hay varias maneras de calcular el área de las figuras dadas.
A continuación veremos 3 posibles maneras de resolver el problema.

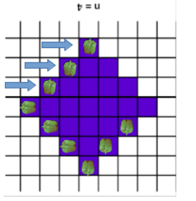
No se proverán demostraciones para las observaciones y formulas que se darán. Sin embargo, todas ellas son demostrables con el uso de inducción y se recomienda al lector realizar dichas demostraciones.

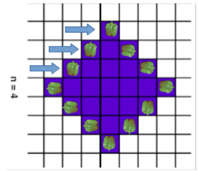
Primera observación: De un (n-1)-polígono a un n-polígono se aumentan $4(n-1)$ cuadrados al polígono.

Ejemplo con 4-polígono de aumento de $4 * 3$ cuadrados:









Área de la figura iv

Como sabemos que el 1-polígono tiene 1 de área podemos ir creando desde ahí el área de los demás:

```
#include <bits/stdc++.h>
using namespace std;
int main(){
  int n, total;
  cin >> n;
  total = 1;
  for(int i = 2; i <= n; i++){
    total += (i-1)*4;
  }
  cout << total << endl;
  return 0;
}
```

Área de la figura v

Lo que hacemos con ese ciclo es sumar:

$$1 + 4(1) + 4(2) + 4(3) + ... + 4(n-1) = 1 + 4\frac{n(n-1)}{2} = 1 + n(n-1)2$$

Área de la figura vi

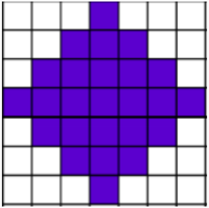
Solución 1:

```
#include <bits/stdc++.h>
using namespace std;

int main(){
  int n;
  cin >> n;
  cout << n*(n-1)*2 + 1 << endl;
  return 0;
}
```

Área de la figura vii

Para la segunda solución, notemos que la figura se encuntra encerrada por un cuadrado de lados $1 + 2(n - 1) = 2n - 1$ (ya que en cada aumento de n-polígono se aumentan dos cuadrados de largo y ancho).



También notemos que para completar cada esquina faltan triángulos de base n-1 (ya que las diagonales de la figura son de tamaño n).

Área de la figura viii

Los triángulos en las esquinas van disminuyendo, de uno en uno, en cada hilera. Lo cual nos permite calcular el total de cuadrados usando la suma de Gauss.

Usando lo anterior y el área del cuadrado, tenemos que el área de la figura es:

$(2n-1)^2 - 4\frac{n(n-1)}{2} = (2n-1)^2 - n(n-1)2$

Área de la figura ix

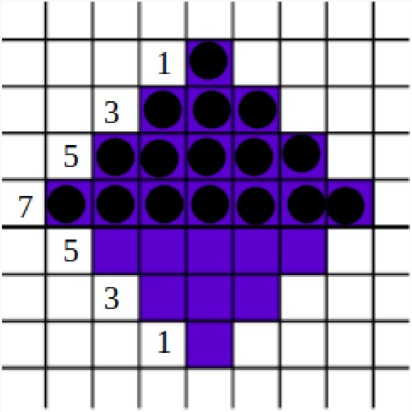
Solución 2:

```
#include <bits/stdc++.h>
using namespace std;

int main(){
  int n;
  cin >> n;
  cout << (2*n-1)*(2*n-1) - (n*(n-1)*2) << endl;
  return 0;
}
```

Área de la figura x

Para la última solución, contemos el número de cuadros de cada fila de la figura.



Área de la figura xi

Notemos que son numeros impares que van disminuyendo desde la franja más ancha.

La siguiente formula nos será de mucha ayuda:
$1 + 3 + 5 + 7 + 9 + ... + 2n - 1 = n^2$

Nuestra figura consta de dos partes:

Una que es de área: $1 + 3 + 5 + 7 + 9 + ... + 2n - 1 = n^2$
Y otra que es de área: $1 + 3 + 5 + 7 + 9 + ... + 2(n-1) - 1 = (n-1)^2$

Con eso podemos ver que el área de la figura es: $n^2 + (n-1)^2$

Solución 3:

```
#include <bits/stdc++.h>
using namespace std;

int main(){
  int n;
  cin >> n;
  cout << n*n + (n-1)*(n-1) << endl;
  return 0;
}
```

Hasta la próxima ☺