

Instituto Universitario de Tecnología para la Informática
I.U.T.E.P.I.
Acarigua – Portuguesa

Guía de Programación III
Estructura Repetitiva FOR en C++
Contadores y Acumuladores.

Programación II
Docente: Ing. Egleé Rivas.

La sintaxis de la estructura repetitiva For es la siguiente:

For(n=1;n<=nºmaxpersonas;n++)

{

Instrucciones de entrada;

Procesos (Estructura selectivas, operaciones algebraicas, contadores, Acumuladores...)

Salidas (las que se amariten)

}

Dónde:

n: representa una variable que controla el ciclo For.

nºmaxpersonas: es la cantidad de personas a las cuales se realiza el ejercicio.

n++: se utiliza para que se incremente el ciclo de uno en uno y se rompa el ciclo cuando llega al nºmaxpersonas.

Se debe declarar la variable n como entera.

Contadores:

Son instrucciones que permiten contar la cantidad de personas involucradas en alguna actividad, su sintaxis es:

Cont++; esto permite contar e ir sumando cada vez que es ejecutada la sentencia de donde está colocado

Cuando se necesita conocer la cantidad de personas involucradas en alguna actividad o proceso, se utiliza un contador.

Acumuladores:

Son instrucciones que permiten sumar y acumular, su sintaxis es:

Acum+= var_acum; donde var_acum es la variable que se va a acumular

Cuando se necesita conocer el total de dinero acumulado en una instrucción o proceso se utiliza un acumulador.

Ejemplo:

De 3 trabajadores se conoce: sueldo, edad, número de hijos. Calcular el salario final de cada trabajador sabiendo que: si la edad es menor de 40 tiene un bono de 3000 bs, le descuentan un 8% del sueldo, en caso contrario el bono es de 4000 bs y el descuento es del 10%; si tiene más de 3 hijos tiene un bono hijo del 30% del sueldo, si el sueldo es menor de 2500 tiene un bono especial del 35% del sueldo. Calcular además:

- Cantidad de empleados con menos de 40 años (contador)
- Cantidad de empleados con bono de 4000 bs (contador)
- Cantidad de empleados con más de 3 hijos (contador)
- Total pagado en bono especial (acumulador)

```
#include<iostream>
#include<stdio.h>
#include<stdlib.h>
#include<conio.h>
#include<iomanip>
using namespace std;
int main()
{
int n=0,cont1=0,cont2=0,cont3=0;
double s,edad,nh,sf,d,bh,be,f,b,acumb;
```

for(n=1;n<=3;n++) (Estructura repetitiva, siempre es igual lo que cambia es el número de personas)

{ (Ésta llave abre el ciclo repetitivo)

```
cout<<"ingrese salario: "<<endl;
cin>>s;
cout<<"ingrese edad: " <<endl;
cin>>edad;
cout<<"ingrese nh: "<<endl;
cin>>nh;
if(edad<=40)
{
b=3000;
d=s*0.08;
cont1++;
}
else
```

Al ejecutar el programa, las entradas son solicitadas en este caso 3 veces porque son 3 empleados, e imprime el sueldo final de cada empleado.

```

{
d=s*0.10;
b=4000;
cont2++;
}
if(nh>=3)
{
bh=s*0.30;
cont3++;
}
if(s<=2500)
{

be=s*0.35;
acumb+=be;
}
sf=s+as+b+be-d;
cout<<"el sueldo final es: "<<fixed<<setprecision(2)<<sf<<endl; (Se imprime
    dentro del ciclo porque es para cada empleado)
} (Ésta llave cierra el ciclo repetitivo, todo lo que está dentro de este ciclo se
    repite tantas veces el enunciado indique cuantas personas sean)
cout<<"cantidad de empleados con menos de 40: "<<cont1<<endl;
cout<<"cantidad de empleados con bono de 600 " <<cont2<<endl;
cout<<"cantidad de empleados con mas de 3 hijos: "<<cont3<<endl;
cout<<"total pago de bono especial: "<<fixed<<setprecision(2)<<acumb<<endl;
}

```

Los contadores y acumuladores se calculan dentro del ciclo pero se imprimen fuera del mismo, como lo vemos en el ejemplo.