

Ciclos de Repetición

Ciclos de Repetición

- Los ciclos de Repetición nos ayudan a simplificar sentencias que deberíamos hacer de forma repetitiva en el código, de esta forma podemos controlar mediante condiciones cuando el ciclo de repetición debe parar y así optimizar y mejorar nuestro código.
- En el ejemplo de arreglos vimos como se llenaban un arreglo de una y dos dimensiones, sin embargo esta forma fue muy repetitiva y muy lenta de hacer, sobre todo cuando los arreglos tienen mas dimensiones y son de mayor tamaño. Gracias a los ciclos de repetición podemos hacer esta y otras sentencias repetitivas mas fáciles a la hora de codificarlas en nuestro algoritmo

Ciclos de Repetición

- Ciclo de repetición “for” (para): Este ciclo de repetición nos permite ejecutar una sección del código de forma repetitiva y contada. El “for” consiste en un contador que va incrementando o disminuyendo desde un punto el cual puede ser cero u otro numero, hasta a otro numero. El incremento o disminución puede ser de uno en uno, dos en dos o lo que deseemos, solo tenemos que tomar en cuenta que debemos especificar hasta que numero debe llegar y que este incremento o disminución deba coincidir con el numero de llegada.
- Un ejemplo mas claro de esto podría ser cuando contarnos del 1 al 10 incrementando 1. El numero de partida es 1, el cual se le debe ir sumando uno en uno hasta llegar a 10. Sin embargo si cambiamos el incremento de 1 por uno de 3 en 3, desde el punto de partida 1 no podremos llegar hasta el numero 10 porque llegado al 9 saltaríamos al 12. Por esto, para poder declarar un “for” se debe tener en cuenta la consistencia de la declaración, sino tendremos un error semántico y nuestro código no funcionara correctamente.

Ciclos de Repetición

- Para poder declarar nuestro “for” , solo hace falta escribir “for” seguidamente se deberá abrir paréntesis y en el escribir primeramente una variable entera a la cual se le deberá ser asignada un valor de partida, es posible declararla en otra parte del código fuera de este paréntesis, pero lo usual es que se declare dentro de ella para que cuando termine el “for”, esta variable no ocupe memoria innecesariamente. Seguidamente de esto, sin cerrar paréntesis se debe escribir “;” para luego escribir la condición de parada que usualmente es la comparación de la variable con el numero de llegada la cual dependerá si la variable esta incrementando o descendiendo. Luego de esto aun sin cerrar paréntesis, se vuelve a poner “;” y seguido de esto se describirá como nuestra variable deberá incrementar, Para esto deberemos escribir la variable de forma que se le asigne la cantidad que actualmente tiene mas lo que se desee incrementar. Finalmente, se cierra paréntesis y se abren llaves en las cuales se declarara las sentencias que se deseen que se repitan tantas veces el “for” haya sido programado.

Ciclos de Repetición

- Ejemplo : Desarrolle un código que sume un entero del 1 al 10 y lo vaya imprimiendo por consola

Ciclos de Repetición

```
C main.c x
1  #include<stdio.h>
2
3  int main() {
4
5      for (int i=1; i<=10; i++){
6          printf("%d\n",i);
7      }
8
9
10
11  return 0;
12
13  }
14
```

Ciclos de Repetición

- Los “for” son una excelente estructura de repetición para llenar arreglos, pues podemos iterar mediante la variable que utilizamos de condición en los arreglos. Por ejemplo llenemos un arreglo de unidimensional de tamaño 10 contando del 10 al 1 disminuyendo 1 e imprimirlo en consola.

Ciclos de Repetición

```
C main.c x
1  #include<stdio.h>
2
3  int main() {
4
5      int a[10];
6
7      for (int i=10; i>0; i--){
8
9          a[i]=i;
10         printf ("%d\n",a[i]);
11     }
12
13     return 0;
14
15 }
16
```


Ciclos de Repetición

- También es posible anidar tantos “for’s” se necesiten dentro de otros “for’s”. Por ejemplo, llenemos un arreglo de tres dimensiones de tamaño 3 cada dimensión mediante “for’s” guardando en ellos la suma de sus posiciones en cada dimensión e imprima en consola lo que se esta guardando de manera ordenada para verse.

Ciclos de Repetición

```
C main.c x
1  #include<stdio.h>
2
3  int main() {
4
5      int a[3][3][3];
6
7      for (int i=0; i<3; i++){
8
9          for(int j=0; j<3; j++){
10
11              for(int k=0; k<3; k++){
12
13                  a[i][j][k]= i+j+k ;
14                  printf(" %d",a[i][j][k]);
15
16              }
17
18              printf("\n");
19
20          }
21
22          printf("\n");
23
24      }
25
26      return 0;
27
28  }
29
```

Ciclos de Repetición

- Ciclo de Repetición “while” (mientras) : esta forma de repetición nos permite ejecutar una pieza de código tantas veces sea necesaria para cambiar una condición que por lo general, el código dentro de este bucle la cambia haciéndola falsa y por consiguiente salirse del bucle, pero también se puede salir del bucle poniendo la sentencia “break” dentro del “while”.
- Para poder ejecutar esta instrucción solo hace falta escribir la frase “while” seguida de paréntesis y escribiendo la condición que se necesita para que esta sección se ejecute, luego de esto se cierra paréntesis y se abren llaves para poner dentro de ellas la sección de código que necesitamos que se repita. Sin embargo, si no es posible parar esta condición estaremos ante otro problema semántico que deberemos resolver, pues nuestro código no funcionaria correctamente

Ciclos de Repetición

- Ejemplo : Desarrolle un algoritmo que le permita al usuario introducir solo números del 1 al 10.

Ciclos de Repetición

```
C main.c x
1  #include<stdio.h>
2
3  int main() {
4
5      int a=13;
6
7      printf("introduzca un numero del 1 al 10 \n");
8      scanf("%d",&a);
9
10     while (!((a>0)&&(a<=10))) {
11
12         printf("Error, debe introducir un numero del 1 al 10 \n");
13         scanf("%d",&a);
14
15     }
16
17
18     return 0;
19
20 }
21
```

Ciclos de Repetición

- Ciclo de repetición Do while: Este tipo de estructura de repetición nos permitirá hacer lo mismo que la estructura “while”, solo que con la particularidad que la sección del código dentro del “do while” se ejecutara primero sin tomar en cuenta la condición de parada, la cual se tomara en cuenta después en el segundo ciclo.
- Para poder ser utilizada solo hace falta poner primero la frase “do” luego abrir llaves, y escribir toda la sección de código que se desea repetir, finalmente se cierran llaves y se escribe “while” seguida de paréntesis en los cuales se escribirá la condición y luego de cerrarlos se escribirá “;”.

Ciclos de Repetición

- Ejemplo: desarrolle un programa en el que si el usuario no introduce l volverá a aparecer el mismo aviso por consola y no podrá cerrarse hasta que lo introduzca.

Ciclos de Repetición

```
C main.c x
1  #include<stdio.h>
2
3  int main() {
4
5      int a=0;
6
7      do {
8
9          printf("Introduzca el numero 0, de lo contrario este aviso volvera a aparecer \n");
10         scanf("%d",&a);
11
12     } while (a!=0);
13
14
15     return 0;
16
17 }
18
```