

Introducción a la programación en C

Informática I - Instituto Unversitario Areonáutico

May 3, 2021

Estructura repetitiva while

Es una estructura repetitiva controlada por centinela. Es decir, que las acciones dentro del bloque de repetición serán ejecutadas **si y sólo si** la expresión lógica evaluada en la condición de ciclo es verdadera.

Estructura while: pseudocódigo

Es una estructura repetitiva controlada por centinela. Es decir, que las acciones dentro del bloque de repetición serán ejecutadas **si y sólo si** la expresión lógica evaluada en la condición de ciclo es verdadera.

Inicio del algoritmo

Estructura while: pseudocódigo

Es una estructura repetitiva controlada por centinela. Es decir, que las acciones dentro del bloque de repetición serán ejecutadas **si y sólo si** la expresión lógica evaluada en la condición de ciclo es verdadera.

Inicio del algoritmo

Mientras (condición de ciclo==Verdadera)

Estructura while: pseudocódigo

Es una estructura repetitiva controlada por centinela. Es decir, que las acciones dentro del bloque de repetición serán ejecutadas **si y sólo si** la expresión lógica evaluada en la condición de ciclo es verdadera.

Inicio del algoritmo

```
Mientras (condición de ciclo==Verdadera)  
    acción 1;
```

Estructura while: pseudocódigo

Es una estructura repetitiva controlada por centinela. Es decir, que las acciones dentro del bloque de repetición serán ejecutadas **si y sólo si** la expresión lógica evaluada en la condición de ciclo es verdadera.

Inicio del algoritmo

Mientras (condición de ciclo==Verdadera)

 acción 1;

 acción 2;

Estructura while: pseudocódigo

Es una estructura repetitiva controlada por centinela. Es decir, que las acciones dentro del bloque de repetición serán ejecutadas **si y sólo si** la expresión lógica evaluada en la condición de ciclo es verdadera.

Inicio del algoritmo

Mientras (condición de ciclo==Verdadera)

 acción 1;

 acción 2;

 acción 3;

Estructura while: pseudocódigo

Es una estructura repetitiva controlada por centinela. Es decir, que las acciones dentro del bloque de repetición serán ejecutadas **si y sólo si** la expresión lógica evaluada en la condición de ciclo es verdadera.

Inicio del algoritmo

Mientras (condición de ciclo==Verdadera)

 acción 1;

 acción 2;

 acción 3;

...

Estructura while: pseudocódigo

Es una estructura repetitiva controlada por centinela. Es decir, que las acciones dentro del bloque de repetición serán ejecutadas **si y sólo si** la expresión lógica evaluada en la condición de ciclo es verdadera.

Inicio del algoritmo

Mientras (condición de ciclo==Verdadera)

acción 1;

acción 2;

acción 3;

...

acción n;

Fin del algoritmo


```
1 #include <stdio.h>
2 int main()
3 {
4     while (numero != -1)
5     {
6         printf("Ingrese un numero entero\n");
7         printf("Para salir ingrese -1\n");
8         scanf("%d",&numero);
9     }
10    return (0);
11 }
```

Estructura repetitiva do-while

La instrucción do...while es controlado por centinela pero evalúa la condición de continuación de ciclo **después de ejecutar el cuerpo del ciclo**; por lo tanto, el cuerpo del ciclo siempre se ejecutará al menos una vez.

Estructura repetitiva do-while: pseudocódigo

La instrucción do...while es controlado por centinela pero evalúa la condición de continuación de ciclo **después de ejecutar el cuerpo del ciclo**; por lo tanto, el cuerpo del ciclo siempre se ejecutará al menos una vez.

Inicio del algoritmo

Hacer

Estructura repetitiva do-while: pseudocódigo

La instrucción do...while es controlado por centinela pero evalúa la condición de continuación de ciclo **después de ejecutar el cuerpo del ciclo**; por lo tanto, el cuerpo del ciclo siempre se ejecutará al menos una vez.

Inicio del algoritmo

Hacer

Estructura repetitiva do-while: pseudocódigo

La instrucción do...while es controlado por centinela pero evalúa la condición de continuación de ciclo **después de ejecutar el cuerpo del ciclo**; por lo tanto, el cuerpo del ciclo siempre se ejecutará al menos una vez.

Inicio del algoritmo

Hacer

acción 1;

Estructura repetitiva do-while: pseudocódigo

La instrucción do...while es controlado por centinela pero evalúa la condición de continuación de ciclo **después de ejecutar el cuerpo del ciclo**; por lo tanto, el cuerpo del ciclo siempre se ejecutará al menos una vez.

Inicio del algoritmo

Hacer

acción 1;

acción 2;

Estructura repetitiva do-while: pseudocódigo

La instrucción do...while es controlado por centinela pero evalúa la condición de continuación de ciclo **después de ejecutar el cuerpo del ciclo**; por lo tanto, el cuerpo del ciclo siempre se ejecutará al menos una vez.

Inicio del algoritmo

Hacer

acción 1;
acción 2;
acción 3;

Estructura repetitiva do-while: pseudocódigo

La instrucción do...while es controlado por centinela pero evalúa la condición de continuación de ciclo **después de ejecutar el cuerpo del ciclo**; por lo tanto, el cuerpo del ciclo siempre se ejecutará al menos una vez.

Inicio del algoritmo

Hacer

acción 1;

acción 2;

acción 3;

...

Estructura repetitiva do-while: pseudocódigo

La instrucción do...while es controlado por centinela pero evalúa la condición de continuación de ciclo **después de ejecutar el cuerpo del ciclo**; por lo tanto, el cuerpo del ciclo siempre se ejecutará al menos una vez.

Inicio del algoritmo

Hacer

acción 1;

acción 2;

acción 3;

...

acción n;

Estructura repetitiva do-while: pseudocódigo

La instrucción do...while es controlado por centinela pero evalúa la condición de continuación de ciclo **después de ejecutar el cuerpo del ciclo**; por lo tanto, el cuerpo del ciclo siempre se ejecutará al menos una vez.

Inicio del algoritmo

Hacer

acción 1;

acción 2;

acción 3;

...

acción n;

Mientras (condición de ciclo==Verdadera)

Estructura repetitiva do-while: pseudocódigo

La instrucción do...while es controlado por centinela pero evalúa la condición de continuación de ciclo **después de ejecutar el cuerpo del ciclo**; por lo tanto, el cuerpo del ciclo siempre se ejecutará al menos una vez.

Inicio del algoritmo

Hacer

acción 1;

acción 2;

acción 3;

...

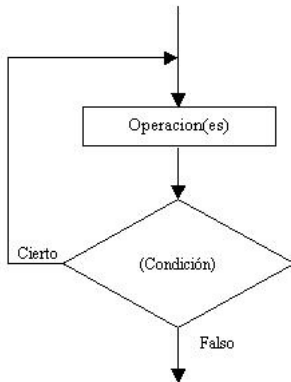
acción n;

Mientras (condición de ciclo==Verdadera)

Fin del algoritmo

Estructura repetitiva do-while: diagrama de flujo

La instrucción do...while es controlado por centinela pero evalúa la condición de continuación de ciclo **después de ejecutar el cuerpo del ciclo**; por lo tanto, el cuerpo del ciclo siempre se ejecutará al menos una vez.



```
1 #include <stdio.h>
2 int main()
3 {
4     int numero=0;
5     do
6     {
7         printf("Ingrese un numero entero\n");
8         printf("Para salir ingrese -1\n");
9         scanf("%d",&numero);
10    } while(numero != -1);
11    return (0);
12 }
```


Estructuras while y do-while: ejemplos I

- 1 Diseñar y codificar un programa que cuente la cantidad de números enteros ingresados por teclado. Cuando el operador ingresa el número "-1" se debe dejar de contar e imprimir el resultado.
[▶ Ver en github](#)
- 2 Diseñar y codificar un programa que imprima todos los números enteros comprendidos entre 0 y N. N debe ser ingresado por el operador. Nota: se debe usar una estructura repetitiva controlada por centinela. [▶ Ver en github](#)
- 3 Modificar el programa anterior, para que imprima todos los números enteros comprendidos entre N_1 y N_2 ambos recibidos por teclado. [▶ Ver en github](#)

Estructuras while y do-while: ejemplos II

- 4 Diseñar y codificar un programa que imprima la suma de todos los números positivos ingresados por teclado. Al ingresar un número negativo, el programa debe imprimir el valor de la suma. [▶ Ver en github](#)
- 5 Diseñar y codificar un programa que permita calcular el promedio de notas de un examen parcial de un curso. El programa debe tomar notas hasta que el operador ingrese la calificación -10.

[▶ Ver en github](#)

Condiciones infinitas, sentencias break y continue

Condiciones infinitas

Cuando en un ciclo repetitivo controlado por centinela la condición a evaluar es siempre verdadera, por ejemplo $1=1$ o $2!=0$, estamos en presencia de un ciclo de repetición infinito.

Sentencia break

Cuando una sentencia break se encuentra dentro de un bucle, este termina su ejecución y el programa continua con las líneas de código que sucedan al ciclo repetitivo.

Sentencia continue

Cuando una sentencia continue se encuentra dentro de un ciclo repetitivo, el programa regresa al comienzo del mismo, evitando todas las líneas que se encuentren después de haber encontrado la sentencia mencionada.

Sentencias break y continue: ejemplos

- 1 Diseñar y codificar un programa que calcule la suma de 10 números ingresados por teclado. Si el operador ingresa algún número par, el programa finalizar la toma de datos e imprimir la suma.

► [Ver en github](#)

- 2 Diseñar y codificar un programa que calcule la suma de 10 números ingresados por teclado. Los números pares no deben ser contemplados en la suma.

► [Ver en github](#)

- 3 Diseñar y codificar un programa que calcule la suma de todos los números ingresados por teclado hasta que el usuario ingresa la opción -10. Además, el numero 100 no debe ser contemplado en la suma total.

► [Ver en github](#)

Repaso general primera parte