Niñas PRO(grama doras)

Niñas PRO(grama doras)

Temario

- ★ Repaso Ciclo For
- ★ Uso de Ciclos While en C++
- ★ Ejemplos de uso de ciclos al leer datos
- ★ Instrucciones continue y break

¿Qué vimos la clase pasada?

Los ciclos nos ayudan a repetir instrucciones sin la necesidad de escribirlas una y otra vez.

La instrucción **for** es útil cuando sabemos el número de repeticiones que debemos realizar o cuando se puede identificar claramente un patrón.

for

```
for (int i=valorInicial; i<=valorFinal; i=i+paso)
{
     ...
     bloque de instrucciones
     ...
}
```

Ejemplo: Números Pares del 1 al 100


```
int main() {
    for (int i=  ?  ; i <=  ?  ; i=  ?  )
    {
        cout << i << endl;
    }
    return 0;
}
```

Ejemplo: Números Pares del 1 al 100


```
int main() {
    for (int i= 2 ; i <= 100 ; i= i+2 )
    {
        cout << i << endl;
    }
    return 0;
}
```

Niñas PRO(grama doras)

Instrucción While

Niñas PRO(grama doras)

Ciclos

Los ciclos son elementos esenciales en cualquier lenguaje de programación.

- El computador ejecuta los algoritmos siguiendo las instrucciones en orden.
- Cuando usamos un ciclo, le decimos al computador que repita las instrucciones que ponemos dentro.
- Los ciclos son útiles porque los computadores pueden ejecutar la misma instrucción muchas veces sin equivocarse.

Ciclos

Antes vimos que el ciclo **for** nos permitía hacer algo un cierto número de veces, pero **¿Qué pasa si no sabemos cuántas veces queremos que se repita una acción?**

Me dicen que juegue a saltar la cuerda y que cuente cuántos saltos puedo hacer. Esto parece un ciclo, pero ¿cómo saber cuántas veces repetir la instrucción?

Ciclo While

★ El ciclo while repite instrucciones mientras la condición sea verdadera, cuando la condición deja de ser verdadera, el ciclo se termina.

```
while (condición) {
    ...
    Bloque de instrucciones
    ...
}
```

Niñas PRO(grama doras)

Ejemplo:

Programemos cómo sería el caso de saltar la cuerda y contar los saltos dió el jugador.

Ejemplo:

Programemos cómo sería el caso de saltar la cuerda y contar los saltos dió el jugador.

```
1.  while ( jugador no ha perdido ){
2.      Dar una vuelta a la cuerda
3.      Si (logró saltar)
4.          incrementar la cantidad de saltos
5.      Sino
6.          perdió
7.  }
8.  Mostrar la cantidad de saltos
```

Ejemplo:

Programemos cómo sería el caso de saltar la cuerda y contar los saltos dió el jugador.

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
      int cantidadDeSaltos = 0;
      bool jugadorPerdió = false;
      while ( jugadorPerdió == false ){
            string salto;
            cout >> ¿Lograste saltar? << endl;
            cin >> salto;
            if (salto == “si”) {
                  cantidadDeSaltos++;
            }
            else {
                  jugadorPerdió = true;
            }
      }
      cout << “Saltos: ” << cantidadDeSaltos << endl;
}
```

Niñas PRO(grama doras)

Lectura de Datos

Usando ciclos

Problema de la contraseña

Niñas PRO(grama doras)

Escribe un programa que se mantenga leyendo una contraseña hasta que ésta sea válida. Cuando la contraseña sea inválida imprime “Senha Invalida”. Cuando la contraseña sea válida imprime “Acesso Permitido” y termina el programa. La contraseña válida es el número 2002.

Entrada	Salida
2003 2022 2002	Senha Invalida Senha Invalida Acesso Permitido

Problema de la contraseña

Niñas PRO(grama doras)

- ¿Qué cosas se **repiten** en la solución de este problema?
- ¿Sabes de antemano cuántas repeticiones debes hacer?
- ¿Qué cosas se realizan **una sola vez** en la solución de este problema?
- ¿Cuál es la **condición** que se debe verificar en cada iteración?

Problema de la contraseña

★ ¿Qué cosas se **repiten** en la solución de este problema?

Se repite la acción de verificar si la contraseña es correcta o no, imprimir si es incorrecta y leer un nuevo intento de contraseña.

★ ¿Sabes de antemano cuántas repeticiones debes hacer?

No, solo sé que debo parar cuando ingresen la contraseña correcta.

Problema de la contraseña

★ ¿Qué cosas se realizan **una sola vez** en la solución de este problema?

La instrucción de escribir "Acesso Permitido" se realiza una sola vez al final.

★ ¿Cuál es la **condición** que se debe verificar en cada iteración?

En cada iteración hay que verificar si el valor ingresado es igual a 2002 o no.

Problema de la contraseña

Veamos la solución en pseudocódigo.

Intenta resolver este problema ingresando a https://www.urionlinejudge.com.br/judge/es/problems/view/1114

```
int main()
{
    Declarar variable para guardar el intento de contraseña
    Leer el primer intento de contraseña

    while ( contraseña distinta a 2002 ) {
        Imprimir mensaje de error
        Leer siguiente intento de contraseña
    }

    Imprimir mensaje de éxito

    return 0;
}
```

Niñas PRO(grama doras)

Instrucciones

Para manipular el flujo de un ciclo

Instrucción `continue`

A veces queremos saltar una iteración, porque para ese caso particular no es necesario realizar todas las instrucciones del ciclo.

La instrucción **`continue`** detiene la iteración en la que está y pasa a la siguiente.

Ejemplo:

```
for(int i = 1; i<=10; i++)
{
     if(i%3==0)
     {
          continue;
     }
     cout << i << endl;
}
```

Instrucción `continue`

A veces queremos saltar una iteración, porque para ese caso particular no es necesario realizar todas las instrucciones del ciclo.

La instrucción **`continue`** detiene la iteración en la que está y pasa a la siguiente.

Ejemplo:

```
for(int i = 1; i<=10; i++)
{
     if(i%3==0)
     {
          continue;
     }
     cout << i << endl;
}
```

1
2
4
5
7
8
10

Instrucción break

A veces queremos detener el ciclo completo sin importar en qué iteración se encuentre.

La instrucción **break** detiene el ciclo al instante, sin importar en qué iteración se encuentre.

Ejemplo:

```
for(int i = 1; i<10; i++)
{
     if(i%3==0){
          break;
     }
     cout << i << endl;
}
```

Instrucción break

A veces queremos detener el ciclo completo sin importar en qué iteración se encuentre.

La instrucción **break** detiene el ciclo al instante, sin importar en qué iteración se encuentre.

Ejemplo:

```
for(int i = 1; i<10; i++)
{
     if(i%3==0){
          break;
     }
     cout << i << endl;
}
```

Línea Gráfica

Lorena Gonzalez - Diseñadora
@soygonzaleztambien