Laços de repetição

Introdução à Prograpação

29 de Março de 2018

Conteúdo da Apresentação

Estruturas de Repetição

- A linguagem C possui os seguintes tipos de estruturas de repetição:
  - *for*
  - *while*
  - *do ... while*

for

- A estrutura do *for* é a seguinte:

for(*inicialização*; *condição de parada*; *incremento*)
{
    código
}

Inicialização

- É somente executado somente uma vez, logo no **início** do **for**.
- É utilizado para inicializar as variáveis que serão utilizadas no loop.
- Exemplo:

```
int i;

for(i = 0; ... ; ...)
{
    ...
}
```

Condição de Parada

- É executada no **final** de cada iteração.
- Se seu valor é verdadeiro (qualquer valor diferente de 0), o **for** é finalizado.
- Caso contrário, a próxima iteração continua normalmente.
- Exemplo:

```
int i;

for(... ; i < 10 ; ...)
{
    ...
}
```

Incremento

- É executada no **final** de cada iteração, **após** a condição de parada ser verificada.
- Usado para alterar o valor de variáveis após cada iteração.
- Exemplo:

```
int i;

for(... ; ... ; ++i)
{
    ...
}
```

Exemplos

- Exemplos de loops **válidos**:

```
for(i = 0 ; i < 10 ; ++i)

for(i = 0; i > -10; --i)

for(i = 2; i < 100; i *= 2)

for(; i < 100; ++i)

for(;;)     // Loop infinito

for(i = 0; ; i++)  // Outro loop infinito
```

Exemplos

- Exemplos de loops **inválidos**:

```
for(i = 0, i < 10 ; ++i)

for(i = 0; i > -10)

for(int i = 2; i < 100; i *= 2)
```

Exercícios

- Encontrar e corrigir os erros do código 'for_1.c'

while

- A estrutura do *while* é a seguinte:

while(*condição de parada*)
{
    código
}

Condição de Parada

- A condição de parada é executada no início de cada loop, ao contrário do **for**.
- Caso a condição seja verdadeira (qualquer valor diferente de 0), o loop é interrompido.
- Exemplo:

```
int i = 0;

while(i < 10)
{
    ...         // O 'i' sera modificado aqui
}
```

Exemplos

- Exemplos de loops **válidos**:

```
while(i < 10)

while(i >= 10)

while(i * i < 100)

while(sin(i) * sqrt(i) + i / 2.0 > 0.3)

while(i += 1, i < 10)

while(1)      // Loop infinito
```

Exemplos

- Exemplos de loops **inválidos**:

```
while(int i = 0)

while()

while(0)  // Loop valido, mas nunca
             sera executado
```

Exercícios

- Encontrar e corrigir os erros do código 'while_1.c'.

while

- A estrutura do *do ... while* é a seguinte:

do
{
    código
}***while***(*condição de parada*);

Condição de Parada

- O funcionamento é igual ao do **while**.
- A única diferença é que mesmo se a condição de parada sempre for falsa, o código dentro do loop será executado pelo menos uma vez
- Exemplo:

```
int i = 0;

do
{
    ...         // Esse codigo sera executado
                // pelo menos uma vez
} while(i < 10);
```

Exemplos

- Exemplos de loops **válidos**:

```
do
{

} while(i < 10);

do
{

} while(i * 2 > -15);

do
{

} while(1);      // Loop infinito
```

Exemplos

- Exemplos de loops **inválidos**:

```
do
{

} while(i < 10)  // Falta o ';'

do while(i * 2 > -15)
{

}

do
{

} while(0);  // Sera executado
             // uma unica vez
```

Exercícios

- Encontrar e corrigir os erros do código 'doWhile_1.c'.

Estruturas de Repetição

- Contar quantos números primos existem até 1.000.
- Contar quantos números primos existem até 10.000.000.
- Dado um raio r, printar uma *círculo* que possua este raio.