

CCP130

Desenvolvimento de Algoritmos

Prof. Danilo H. Perico



Estruturas de Repetição



Repetições

- São utilizadas para executar **várias vezes** a mesma parte do programa
- Normalmente dependem de uma condição
- Repetições são a base de vários programas!



Exemplo

- Fazer um programa para **imprimir 10 números sequenciais** na tela, começando do número 1.



Exemplo

- Esta é uma solução para o problema do exemplo:
- É uma solução boa?!

```
1  #include <stdio.h>
2
3  ~ int main(void){
4      printf("1\n");
5      printf("2\n");
6      printf("3\n");
7      printf("4\n");
8      printf("5\n");
9      printf("6\n");
10     printf("7\n");
11     printf("8\n");
12     printf("9\n");
13     printf("10\n");
14 }
```



Exemplo

- Esta é uma solução para o problema do exemplo:
- É uma solução boa?!
- **Não!**
Muitos comandos repetidos!

```
1  #include <stdio.h>
2
3  ~ int main(void){
4      printf("1\n");
5      printf("2\n");
6      printf("3\n");
7      printf("4\n");
8      printf("5\n");
9      printf("6\n");
10     printf("7\n");
11     printf("8\n");
12     printf("9\n");
13     printf("10\n");
14 }
```



Repetições

- Idealmente, poderíamos ter o comando `printf(x)` imprimindo a variável `x`
 - onde `x` variasse de 1 até 10



Repetições

- Idealmente, poderíamos ter o comando `printf(x)` imprimindo a variável `x`
 - onde `x` variasse de 1 até 10
- A boa notícia é que existem comandos para realizar este tipo de ação!



Comando
while



Comando **while**

- O comando **while** (enquanto) serve para executarmos alguma repetição **enquanto** uma **condição for verdadeira**

```
while(condição)
{
    //instruções dentro do loop;
}
```



Comando **while**

- Chamamos de **iteração*** a repetição de todas as instruções que estão dentro da estrutura de repetição

*** iteração:** ato de iterar**; repetição.

**** iterar:** tornar a fazer; repetir.



Comando *while*

Exemplo 1:

- Imprimir números sequenciais de 1 a 10

```
1  #include <stdio.h>
2
3  int main(void){
4      int i = 1;
5      while(i <= 10){
6          printf("%d\n", i);
7          i++;
8      }
9  }
```



Comando *while*

Exemplo 2:

- Escreva um programa que faça a **contagem regressiva de 10 até 0**
- O programa deve imprimir **cada número da contagem a cada um segundo**

Função **sleep(x)** - pausa o código por **x** segundos

sleep(1) - pausa o código por **1** segundo

```
#include <unistd.h> —————→ sleep(1);
```

Comando *while*

Exemplo 3:

- Impressão do número 1 até um número digitado pelo usuário

```
1  #include <stdio.h>
2
3  ~ int main(void){
4      int i = 1, ultimo;
5      printf("Digite o ultimo numero da contagem:");
6      scanf("%d", &ultimo);
7  ~  while(i <= ultimo){
8          printf("%d\n", i);
9          i = i + 1;
10     }
11 }
```

- O lado direito do sinal de igual (=) é executado primeiro!
- O resultado é atribuído para a variável que estiver do lado esquerdo do sinal de igual (=)



Comando *while*

Exemplo 4: combinando repetição com *if*

- Programa que imprime todos os números pares de 0 até um número digitado pelo usuário.



Comando *while*

Exemplo 5:

- Programa para calcular a somatória de 10 números que devem ser digitados pelo usuário

Comando *while*

```
1  #include <stdio.h>
2
3  int main(void){
4      int i = 0, somatoria = 0, num;
5      while(i < 10){
6          printf("Digite um numero: ");
7          scanf("%d", &num);
8          somatoria = somatoria + num;
9          i = i + 1; //i++
10     }
11     printf("Somatoria: %d\n", somatoria);
12 }
```

somatoria é um
acumulador

i é um contador

um acumulador é uma
variável que guarda uma
soma



Comando **while**

while infinito:

- Muitas vezes, queremos que nossos programas sejam executados **infinitamente**
- Nesses casos, podemos utilizar uma condição que nunca deixe de ser verdadeira



Comando **while**

while infinito:

```
while(1){  
    // código que vai executar infinitamente  
}
```



Comando *break*

- Porém, mesmo quando utilizamos um *while infinito*, é possível que em determinadas situações o programa precise sair do loop de repetição
- Esta interrupção pode ser alcançada com o comando *break*
- O comando *break* pode ser utilizado para *interromper* o while, independentemente da condição



Comando *break*

- Exemplo 6:
 - Somatória de valores digitados pelo usuário até que o número 0 (zero) seja digitado; quando 0 for digitado o resultado da somatória é exibido



Repetições aninhadas

- Podemos combinar vários **while**, um dentro do outro!
- Com isso, conseguimos alterar automaticamente o valor de mais do que somente uma variável



Repetições aninhadas

- Exemplo 7:
 - Gerar automaticamente as Tabuadas de 1 a 10!



Comando *for*



Comando *for*

- O comando *for* (para) serve, também, para executarmos repetições

```
for (valor_inicial; condiçã_final; valor_incremento)
{
    // instruções dentro do loop;
}
```



Comando *for*

- *for* é a estrutura de repetição mais utilizada
- Uma vantagem do *for* com relação ao *while* é que o *contador* não precisa ser explícito!



Comando *for*

Exemplo 8:

```
#include <stdio.h>

int main(void)
{
    int contador;
    for(contador = 1; contador <= 10; contador++)
    {
        printf("%d ", contador);
    }
    return(0);
}
```



Comando *for*

Exemplo 9:

- Faça um programa que gera 100 números aleatórios entre 1 e 100 e, então, exiba qual foi o maior número gerado

```
#include <stdlib.h> —————> rand() % 100;
```

```
#include <time.h> —————> srand(time(NULL));
```

Comando

do ... while



Comando *do - while*

- O comando *do - while* (faça - enquanto) serve, também, para executarmos repetições

```
do{  
    // instruções dentro do loop  
}while(condição)
```



Comando *do - while*

- O comando *do - while* garante que o bloco de instruções seja executado **no mínimo uma vez**, já que a condição que controla o laço é testada apenas no final do comando




Comando *do - while*

Exemplo 10:

- Faça um programa para calcular a média entre 2 notas. No final de cada cálculo o programa deve perguntar ao usuário se ele deseja fazer outro cálculo ou não.

Exercícios

- 
1. Escreva um programa que calcula a média aritmética de 5 números digitados pelo usuário. Utilize contadores e acumuladores.

considerando que o primeiro número digitado é $n1$, o segundo $n2$, ...

$$media = \frac{(n1+n2+n3+n4+n5)}{5}$$