

Pratica 2

Comandos de repetição (Loops)

Todos os Loops têm a mesma função que é repetir uma instrução (ou um conjunto de instruções) num número determinado ou indeterminado de vezes. Na linguagem C existem três comandos de repetição: **while**, **do-while** e **for**. Elas geralmente são usadas quando precisamos executar uma rotina (conjunto de instruções) mais de uma vez. Isso nos ajuda a simplificar o código fonte, deixando-o mais simples de entender. Aqui veremos todos os comandos de repetição que vamos usar durante a matéria.

1. Comando de repetição: while

O comando while permite que um certo trecho de programa seja executado ENQUANTO uma certa condição for verdadeira. A forma do comando while é a seguinte:

```
while (condição) {  
    // comandos a serem repetidos  
    // comandos a serem repetidos  
}  
  
// comandos após o 'while'
```

O funcionamento é o seguinte:

- i. Testa a condição;
- ii. Se a condição for falsa então pula todos os comandos do bloco subordinado ao while e passa a executar os comandos após o bloco do while;
- iii. Se condição for verdadeira então executa cada um dos comandos do bloco subordinado ao while;
- iv. Após executar o último comando do bloco do while volta ao passo 1.

As condições usada no **while** são condições lógicas, assim como no **if/else**. A condição deve retornar um valor verdadeiro ou falso. Abaixo segue um exemplo de código c usando a estrutura de repetição **while**.

```
1  #include <stdio.h>  
2  #include <stdlib.h>  
3  
4  int main()  
5  {  
6      int opcao = 1;  
7  
8      while(opcao != 0) {  
9          printf("Digite a opcao desejada:\n");  
10         printf("0) Sair.\n");  
11         printf("1) continuar.\n");  
12         scanf("%d", &opcao);  
13         system("cls");  
14     }  
15  
16  
17     return 0;  
18 }
```

2. Comando de repetição: do_while

O comando do-while permite que um certo trecho de programa seja executado ENQUANTO uma certa condição for verdadeira. A forma do comando do-while é a seguinte:

```
do {  
    // comandos a serem repetidos  
    // comandos a serem repetidos  
} while (condição);  
// comandos após o 'do-while'
```

O funcionamento é o seguinte:

- i. Executa os comandos dentro do bloco do-while;
- ii. Testa a condição;
- iii. Se a condição for falsa então executa o comando que está logo após o bloco subordinado ao do-while;
- iv. Se condição for verdadeira então volta ao passo 1.

As condições usada no **do_while** são condições lógicas, assim como no **if/else**. A condições deve retornar um valor verdadeiro ou falso. Abaixo segue um exemplo de código c usando a estrutura de repetição **do_while**.

```
1  #include <stdio.h>  
2  #include <stdlib.h>  
3  
4  int main()  
5  {  
6      int opcao;  
7      float numero_1, numero_2;  
8  
9      do {  
10         printf("Digite o primeiro valor:\n");  
11         scanf("%f", &numero_1);  
12         printf("Digite o segundo valor:\n");  
13         scanf("%f", &numero_2);  
14         printf("Resultado da soma: %.2f\n", (numero_1 + numero_2));  
15         printf("Deseja efetuar outra soma? [ 1 (sim) ou 0 (nao)]\n");  
16         scanf("%d", &opcao);  
17         system("cls");  
18     }while(opcao == 1);  
19  
20  
21     return 0;  
22 }
```

3. Comando de repetição: for

O comando **for** permite que um certo trecho de programa seja executado um determinado número de vezes.

A forma do comando **for** é a seguinte:

```
for (comandos de inicialização; condição de teste; incremento/decremento) {  
    // comandos a serem repetidos  
    // comandos a serem repetidos  
}  
  
// comandos após o 'for'
```

O funcionamento é o seguinte:

- i. Executa os comandos de inicialização;
- ii. Testa a condição;
- iii. Se a condição for falsa então executa o comando que está logo após o bloco subordinado ao for;
- iv. Se condição for verdadeira então executa os comandos que estão subordinados ao for;
- v. Executa os comandos de incremento/decremento;
- vi. Volta ao passo 2.

O comando **for** possui 3 parâmetros. O primeiro é onde definimos a inicialização do contador, o segundo definimos a condição de parada (uma condição logica assim como no **if/else**) e o terceiro definimos uma regra para alterar o contador. Tirando o contador da condição logica os outros dois não são necessários para o **for**. Abaixo segue um exemplo de código c usando a estrutura de repetição **for**.

```
1  #include <stdio.h>  
2  #include <stdlib.h>  
3  
4  int main()  
5  {  
6      for(int i=0; i<=10; i = i + 2) {  
7          printf("O numero corrente eh: %d\n",i);  
8      }  
9  
10     return 0;  
11 }
```

Exercícios

1. Escrever um programa para exibir os números de 1 até 50 na tela.
2. Fazer um programa para imprimir todos os números pares entre 567 e 7932.
3. Ler um valor inteiro (aceitar somente valores entre 1 e 10) e escrever o resultado da multiplicação do número lido com os valores de 11 a 20.
4. Leia várias idades e calcule a média entre as idades (usar uma variável para idade). Lei no mínimo 10 idades.
5. Ler 10 números e imprimir quantos são pares e quantos são ímpares.
6. Utilizando a estrutura de repetição **for**, faça um programa em C que receba 10 números e conte quantos deles estão no intervalo [10,20] e quantos deles estão fora do intervalo, escrevendo estas informações.
7. Crie um programa que imprima 11 vezes a frase " Hello World!" utilizando uma estrutura sequencial e uma estrutura de repetição **while**.
8. Lei um valor inteiro do usuário. Enquanto o valor for par imprime o valor, quando o valor for ímpar pare de pedir números para o usuário.
9. Crie um programa para ler 3 notas e mostrar a média delas. Você só pode usar duas variáveis, uma como contador e outra para calcular a média.
10. Leia vários números inteiros seguidos, multiplique todos os valores lidos. No momento que o usuário digitar 0 (zero) pare de ler números e mostre o resultado da multiplicação na tela. O valor zero não deve ser usado na multiplicação. Caso o primeiro valor digitado pelo usuário for zero, alerte o usuário que ele finalizou o programa sem digitar nenhum número.
11. Na matemática, a sucessão de Fibonacci (ou sequência de Fibonacci), é uma sequência de números inteiros, começando normalmente por 0 e 1, na qual cada termo subsequente corresponde à soma dos dois anteriores. Em termos matemáticos, a sequência é definida recursivamente pela fórmula abaixo:

$$F_n = F_{n-1} + F_{n-2};$$

e valores iniciais

$$F_1 = 0, F_2 = 1.$$

Lei o valor de “n” e calcule o Fibonacci deste valor. Imprima o resultado. Use o comando de repetição que achar melhor. O usuário não pode digitar valores menores que 1.