

Aula 07

MC 102 - Algoritmos e Programação de Computadores

Comandos de repetição I: *while* e *do while*.

Repetição

Até agora, aprendemos como escrever programas capazes de executar comandos de forma linear (seqüenciadamente), tomando decisões sempre que necessário antes de executar um comando, ou um bloco deles.

Mas se precisar repetir algum bloco de comando, devo “copiar e colar” sempre?

Repetição

Exemplo: imprimir todos os números de 1 a 5.

```
...  
printf("1");  
printf("2");  
printf("3");  
printf("4");  
printf("5");
```

Repetição

E para imprimir todos os números de 1 a 100?

```
printf("1");  
printf("2");  
..  
printf("98");  
printf("99");  
printf("100");
```

Repetição

E para imprimir todos os números de 1 a n?

```
...
printf("1");
if (n >= 2)
    printf("2");
if (n >= 3)
    printf("3");
...
```

Não tem nada melhor?

Comando while

É usado para a execução repetida de um bloco de comandos, os quais comandos serão executados enquanto uma expressão lógica for verdadeira.

```
while (<expressão_lógica>) {  
    comandos;  
    ...  
}
```

Comando while

Imprima todos os números de 1 a 100.

```
int i;
i = 1;
while (i < 100) {
    printf("%d\n", i);
    i++;
}
```

Comando while

Imprima todos os números de 1 a 100.

```
int i;
i = 1;
while (i < 100) {
    printf("%d\n", i);
    i++;
}
```

Quando a expressão
será falsa?

Atenção!!!

Se as **variáveis** da expressão_lógica **nunca** se **alterarem**, o programa **nunca sai do loop** while e pode caracterizar um **loop infinito!** É preciso que alguma **variável** tenha seu valor alterado dentro do loop while.

```
int i;  
  
i = 1;  
  
while (i < 100) {  
    printf("%d\n", i);  
}
```

O que aconteceria aqui?

Caso 1 – Soma

Como ler n números inteiros e retornar a sua soma?

Algoritmo

Informe um número n

Soma = 0

Enquanto n for maior que 0

Informe um número x

Soma = Soma + x

n = n - 1

Imprima Soma

Caso 1 – Soma

Como ler n números inteiros e retornar a sua soma ?

Programa

```
#include <stdio.h>

int main(int argc, char **argv) {
    int n, x, soma;

    printf("Informe quantos numero deseja somar: ");
    scanf("%d", &n);
    soma = 0;
    printf("Informe os numeros linha a linha:\n");
    while (n > 0) {
        scanf("%d", &x);
        soma += x;
        n--;
    }
    printf("\nSoma = %d\n", soma);
}
```

Caso 2 – MDC

Calcular o Máximo Divisor Comum (MDC) entre dois números inteiros positivos

Algoritmo

Informe um número a e b

Enquanto ($b > 0$)

 resto = $a \% b$

 a = b

 b = resto

Imprima a

Caso 2 – MDC

Calcular o Máximo Divisor Comum (MDC) entre dois números inteiros positivos

Programa

```
#include <stdio.h>

int main(int argc, char **argv) {
    unsigned int a, b, resto;

    printf("Informe a e b inteiros positivos: ");
    scanf("%u %u", &a, &b);

    while (b > 0) {
        resto = a % b;
        a = b;
        b = resto;
    }

    printf("MDC = %u\n", a);
}
```

Caso 3 – Fatorial

Calcular o número fatorial de n p/ $n \leq 20$.

Algoritmo

Informe o número n

Se ($n > 20$)

O valor máximo para n é 20.

n = 20

fat = 1

i = n

Enquanto ($i > 1$)

fat *= i

i--

Imprima fatorial(n) = fat

Caso 3 – Fatorial

```
#include <stdio.h>

int main(int argc, char **argv) {
    unsigned short n, i;
    unsigned int fat;

    printf("Informe um inteiro positivo: ");
    scanf("%u", &n);
    if (n > 20) {
        printf("O valor maximo para n eh 20.\n");
        n = 20;
    }
    fat = 1;
    i = n;
    while (i > 1) {
        fat *= i;
        i--;
    }
    printf("fatorial(%u) = %u\n", n, fat);
}
```

Exercício

Escreva um programa no qual o usuário vai entrando sucessivamente com inteiros positivos. Quando o usuário entrar com um valor negativo o programa pára de pedir valores e calcula a média dos valores já fornecidos.

Exercício

```
#include <stdio.h>

int main(int argc, char **argv) {
    int num, cont;
    float soma, media;
    printf("Digite um numero:");
    scanf("%d", &num);

    soma = 0.0; cont = 0;
    while (num >= 0) {
        soma += num;
        cont++;
        printf("Digite outro numero:");
        scanf("%d", &num);
    }
    media = soma/cont;
    if (cont > 0)
        printf("A média dos %d numeros digitados eh %f\n",
            cont, media);
    else
        printf("Não foram informados números para o cálculo \n");
    return 0;
}
```

Repetição

O que acontece se a condição inicial do loop while for falsa?

Os comandos while não serão executados!

E se eu quiser executar o comando antes de testar a condição ?

Exemplo: Some números inteiros até que o valor digitado seja menor ou igual a zero!

Caso 4 – Repetição

Soma de inteiros positivos

Algoritmo

Soma = 0

Faça

Informe um número x

Se $x > 0$

Soma = Soma + x

Enquanto ($x > 0$)

Se Soma > 0

Imprima Soma

Comando do - while

Um bloco de comandos **SEMPRE** será executado enquanto uma expressão lógica for verdade.

```
do {  
    comandos;  
    ...  
} while (<expressão_lógica>;
```

Cálculo da Média – do while

```
#include <stdio.h>

int main(int argc, char **argv) {
    int num, cont;
    float soma, media;
    printf("Digite um numero:");
    soma = 0.0; cont = 0;
    do {
        scanf("%d", &num);
        if (num >= 0){
            soma += num;
            cont++;
            printf("Digite novo numero:")
        }
    } while (num >= 0);
    media = soma/cont;
    if (cont > 0)
        printf("A média dos %d numeros digitados eh %f\n",
            cont, media);
    else
        printf("Não foram informados números para o cálculo \n");
    return 0;
}
```

Caso 5 – Pirraça!

Programa

```
#include <stdio.h>

int main(int argc, char **argv) {
    char n;
    do {
        printf("\nEste programa não faz nada.\n");
        printf("Digite 0 para sair: ");
        scanf("%d", &n);
    } while (n != 0);
}
```

Exercício – Problema do Troco

Escreva um programa que encontre o menor número possível de moedas para formar uma quantia Q de dinheiro. Esta quantia é um **inteiro positivo** informado em centavos.

Existem apenas as moedas de 1, 5, 10, 25, 50 e 100 centavos. Imprimir o número de moedas necessárias de cada tipo.