

# ***Projeto de Lógica de programação***

Grupo Pet-Computação

*Matheus Freitag e Renata Junges*

# Laços de repetição

- O comando de repetição while:

## Sintaxe da instrução:

```
while (expressão teste){  
  
    //Bloco de instruções  
  
}
```

A instrução **while** é usada pra dizer que o bloco de instruções será executado *enquanto a expressão teste for verdadeira*, ou seja, quando acabar o bloco de instruções deve-se voltar ao comando **while** e testar novamente a condição, se for verdadeira o bloco deve ser executado de novo, e assim sucessivamente.

## Exemplo:

```
#include <stdio.h>
int main(){
    int total = 0, num;

    while ( total < 20 ) {
        printf( "Total = %d\n", total );
        printf( "Entre com um numero: " );
        scanf( "%d", &num );
        total += num;
    }
    printf( "Final total = %d\n", total );
    return 0;
}
```

Este programa pede números ao usuário até que a soma de todos os números digitados for pelo menos 20.

- O comando de repetição do-while:

### Sintaxe da instrução:

```
do{  
    \\Bloco de instruções  
}while (condição);
```

O laço de repetição **do-while**, funciona como o **while**, porém garante que o **bloco de instruções vai ser executado ao menos uma vez**. O programa executará o bloco de instruções e só então testará a condição, se ela for verdadeira ele executa novamente o bloco, se não, pula para a próxima instrução do programa.

## Uso do do-while em menus:

```
#include <stdio.h>

int main (){
    int i;

    do{
        printf ("\nEscolha a fruta
pelo numero:\n");
        printf("\t(1)Mamão\n");
        printf("\t(2)Abacaxi\n");
        printf ("\t(3)Laranja\n");
        scanf("%d", &i);
    } while ((i<1)|| (i>3));
```

```
switch (i){
    case 1:
        printf ("\tVocê escolheu
Mamão.\n");
        break;
    case 2:
        printf ("\tVocê escolheu
Abacaxi.\n");
        break;
    case 3:
        printf ("\tVocê escolheu
Laranja.\n");
        break;
}
return 0;
}
```

# Laços de repetição

- O comando de repetição for:

## Sintaxe da instrução:

```
for (inicialização ; até quando irá ser executado ; incremento ou decremento){  
  
    //Bloco de instruções  
  
}
```

A instrução **for** é usada pra dizer que o bloco de instruções será executado a partir de um determinado valor até outro valor, incrementando ou decrementando esse valor. O bloco de instruções será repetido para todos os valores entre a inicialização e o último valor a ser executado.

## Exemplo:

```
#include <stdio.h>
```

```
int main(){
```

```
    int i;
```

```
    for ( i=0 ; i<5 ; i++) {  
        printf( "%d \n",i );
```

```
    }
```

```
    return 0;
```

```
}
```

Este programa  
escreve na tela os  
números de 0 a 4.

Inicializações:

- i = 0;
- i = 1;
- i = 2;
- i = 3;
- i = 4.

# Exercício de fixação:

1. Faça um programa que mostre na tela todos os números de 1 a 100 usando laço while.
2. Faça um programa que dada uma quantia de dinheiro em libras inglesas imprime o equivalente em dólares canadenses. Utilize a relação  $\text{£}1.0 = \$1.84$ , depois disso, pergunte se o usuário quer ou não transformar mais algum valor; se não quiser, encerre. (Use do-while).
3. Faça um programa que compute quantos são os múltiplos de 2, de 3 e de 5 entre 1 e 100; compute também quantos são os números múltiplos de 2, 3 e 5 ao mesmo tempo. (Use o for)

Observações: implemente um contador para os múltiplos de 2, outro para os de 3 e outro para os de 5. Depois implemente um contador para armazenar a quantidade de números múltiplos de 2, 3 e 5 ao mesmo tempo. Imprima na tela os dados obtidos.



Grupo Pet-Computação

*Matheus Freitag e Renata Junges*