Estrutura de Repetição

Baseada em Condição

Neste material vamos iniciar os estudos das estruturas de repetição. Tais estruturas são divididas em duas classes, as baseadas em: condição e contagem. **Opa!** Se nosso último assunto foi as estruturas condicionais, vamos **começar com essa classe? Bora-a!**

Comecemos pela necessidade:

Imaginemos a necessidade de solicitar a idade de um usuário. Há! Facinho, né?

```
#include <stdio.h>

main( ) {
int idade;
      printf(" Informe a idade: ");
      scanf( "%d",&idade);
}
```

Imagina solicitarmos de dois usuários. Podemos pensar: H'm, é simples! Basta **duplicar:**

```
#include <stdio.h>

main( ) {
int idade1, idade2;
      printf(" Informe a idade: ");
      scanf( "%d",&idade1);

      printf(" Informe a idade: ");
      scanf( "%d",&idade2);

}
```

Imagina mais, imagina solicitarmos de 10 usuários. Iríamos criar 10 variáveis e copiar e colar o par de instruções printf e scanf? Nossa! Que código extenso, ein?

Agora, bora observar outra situação, a situação de solicitar a idade de **n** usuários. **Complicou? Então vamos descomplicar!**

Na linguagem de programação C, há uma instrução que **permite a repetição de um conjunto com uma ou mais linhas de comandos**, repetidas vezes, desde que uma condição sempre seja verdade. Basicamente seria:

enquanto (condição for verdade) { repita uma ou mais instruções }

A essa instrução chamamos de **while**. E sua sintaxe é definida como:

while (<condições>) {<comandos>}

As condições são expressões lógicas avaliadas no início da instrução **while**, que quando verdadeiras, permitem a execução dos comandos a seguir. Quando temos uma ou mais linhas de comandos para repetir é necessário a definição de um bloco **{ }**, quando temos só uma linha de comando a repetir não é obrigado um uso de um bloco **{ }**.

Vamos voltar para aquele nosso exemplo inicial? Bora-a! uhu!

Precisamos solicitar 10 idades. Observem, já temos aí uma condição: tenho que pedir 5 nomes? Não! Tenho que pedir 300 nomes? Não! Temos que pedir 10! Então 10 é nossa quantidade de idades a receber. Logo, é nossa condição de repetição. Até conseguir capturar 10 idades nosso programa não deve párar de pedir idades. Então, vamos esquematizar o que já pensamos:

- Temos uma quantidade;
- Essa quantidade de idades requeridas deve ser nem mais, nem menos que 10;

Ou seja enquanto a quantidade de idades recebidas não for igual a 10, receberemos mais idade, até conseguimos todas as 10. Então nossa instrução poderia ser algo assim:

while (idade <10)

sabendo que idade é um número inteiro e nulo.

Vamos substituir no nosso primeiro código desse documento?

```
#include <stdio.h>

main( ) {
int idade;
int quantidade=0;
        while (quantidade<10)
          printf(" Informe a idade: ");
          scanf( "%d",&idade);
}
```

Tudo certo? Quase certo!

Lembra que falamos sobre definir se iremos repetir um ou mais instruções? Pois é, e nesse caso, queremos repetir quantas instruções 10 vezes? Queremos solicitar e receber 10 idades, então queremos repetir o par printf e scanf. Se queremos repetir mais de uma instrução, vamos definir o bloco de instruções a serem repetidas no while:

```
#include <stdio.h>
main( ) {
int idade;
int quantidade=0;
        while (quantidade<10) {
          printf(" Informe a idade: ");
          scanf( "%d",&idade);
        }
}
```

Tudo tranquilo, por enquanto? Espero que sim! Esse código já tá pronto? 'Cês acham? **Não, né? Muito bem!**

Pois é, observando, em algum momento a variável quantidade muda seu valor? Muda seu valor de forma a ser maior que 10? Não né? Pois é, então uma vez "entrando" nessa instrução do while a condição sempre será verdade. Logo, sempre iremos executar essa repetição. Logo, sempre estaremos em um "looping". Logo, nosso programa não tem fim. Logo, não podemos defini-lo como um algoritmo. Lembra? **Todo algoritmo tem começo, meio e fim**. Bora, consertar? Bora-a!

A proposta é que a cada passo que recebemos uma idade, contabilizemos a quantidade de idades recebidas +1, então vamos contabilizar?

```
#include <stdio.h>
main( ) {
int idade;
int quantidade=0;
        while (quantidade<10) {
          printf(" Informe a idade: ");
          scanf( "%d",&idade);
          quantidade = quantidade +1;
        }
}
```

Agora sim, tudo certo e nada errado?! Ué, mas bora exibir no final desse programa quantas idades recebemos? Vamos:

```
#include <stdio.h>
main( ) {
int idade;
int quantidade=0;
        while (quantidade<10) {
          printf(" Informe a idade: ");
          scanf( "%d",&idade);
          quantidade = quantidade +1;
        }
        printf("quantidade de idades lidas = %d", quantidade);
}
```

Agora sim, fechou!
Tudo entendido sobre o while? Ótimo! Facinho, né? Basta só praticar.
Mas antes, vamos conhecer a outra instrução de repetição baseada em repetição? Prometo que será breve. Vamos lá!

Bom, sempre ao utilizamos o while verificamos se a condição é verdadeira, se sim, um conjunto de instruções será repetidos. Em C temos **outra instrução que realiza pelo menos uma vez as instruções de repetição para depois verificar a necessidade de repetir mais vezes.** Basicamente seria:

faça { repetidas vezes uma ou mais instruções } enquanto (condição for verdade);

A essa instrução chamamos de **do-while**. E sua sintaxe é definida como:

```
do {<comandos>} while (<condições>);
```

Colocando nosso código anterior junto ao código que utiliza **do-while**, veremos:

```
#include <stdio.h>
main( ) {
int idade;
int quantidade=0;
        while (quantidade<10) {
          printf(" Informe a idade: ");
          scanf( "%d",&idade);
          quantidade = quantidade +1;
        }
        printf("quantidade de idades lidas = %d",
        quantidade);
}
```

```
#include <stdio.h>
main( ) {
int idade;
int quantidade=0;
        do{
          printf(" Informe a idade: ");
          scanf( "%d",&idade);
          quantidade = quantidade +1;
        } while (quantidade<10);
        printf("quantidade de idades lidas = %d",
        quantidade);

}
```

Observem, o código é praticamente o mesmo a diferença é que com o **do-while** solicitamos pelo menos uma idade.

Você consegue observar quais atenções o uso do while e do-while poderíamos ter para exemplos como esse? Se sim, passa lá no fórum e compartilha. Estamos esperando sua interação por lá!