Aula 3 – Modulo Iniciante

Recapitulando...

Até agora estudamos:

- Estrutura de um código
- Variáveis e seus tipos
- Operadores Matemáticos, Lógicos, de Incremento, de Decremento e de Comparação
- Printf e Scanf
- If - Else

Laços de Repetição

Conceitos que vocês precisam saber, basicamente, sobre laços de repetição:

1. Todos vão utilizar uma variável, para servir como contador
2. Todo laço de repetição precisa ter uma condição de parada(do contrário ele vai ficar rodando para sempre)
3. Vamos falar nesse resumo na ordem:
   - While
   - For

While

- While : Traduzindo do inglês significa ENQUANTO

- O while é um típico laço de repetição, usado quando você quer que o programa repita enquanto uma condição for verdadeira. Geralmente é usado quando você quer que o programa repita um número indeterminado de vezes.

Exemplo: “Faça um programa que leia um número e printe o valor do número 2 na tela. Pare o programa quando o valor digitado for 0”.

- No exemplo acima, percebam que o programador não sabe quantas vezes o programa tem que repetir, então o programa tem que repetir ENQUANTO(while) o número digitado for diferente de zero

Sintaxe do While

```
while( condição de parada )
{
    // comando 1
    // comando 2
    // comando ...
}
```

Mas afinal, o que posso fazer com while?

- Ainda pensando no problema anterior, para resolve-lo podemos usar uma certa peculiaridade do while colocando um scanf como condição de parada.

```
#include <stdio.h>
main(){
    int a;
    while(scanf("%d", &a)){
        if(a == 0) break;
    }
}
```

- O comando break QUEBRA o laço de repetição. Ele que faria o nosso programa parar quando o usuário digitar zero;

Entao sempre preciso usar um break?

- NÃO, nós também podemos fazer o programa rodar um número determinado de vezes com o while.

- Para isso, nós criamos uma variável, e damos uma condição de parada dentro dos parênteses do while para que o programa não rode infinitamente (entre em loop infinito)

Entao sempre preciso usar um break?

Exemplo: Quero que meu programa repita 100 vezes

```
#include <stdio.h>
main(){
    int x = 0;
    while(x < 100){
        // Comandos
        x++;
    }
}
```

O que esta acontecendo?

- Criamos um contador chamado x, que começa valendo 0, e enquanto x for menor que 100, o while irá repetir.
- Pórem a cada vez que ele repete o x vai ser acrescentado de mais 1 (x++), fazendo com que o programa repita 100 vezes;

For

- O For é outro modelo bastante usado na repetição, que é usado quando você já sabe quantas vezes o seu programa vai repetir.
- Por exemplo, se eu quero ler um número N, e fazer meu programa rodar N vezes, eu uso o for.

- “Mas eu não posso usar o while?”
- Pode, porém, o for é mais indicado, pois a função dele é exatamente repe tir um número determinado de vezes... da menos trabalho pra escrever.

For

```
for(valor inicial do contador; condição de parada; condição de incremento/decremento){
        //comandos
}
```

- No for seu contador sempre vai começar valendo alguma coisa e vai indo em direção a outra coisa... Como assim???
- É como se ele tivesse um ponto de partida, um ponto de chegada e a maneira como ele vai percorrer essa "jornada".

1. O valor inicial é o seu "ponto de partida.
2. A sua condição de parada é o ponto de chegada.
3. E a condição de incremento/decremento é a maneira como ele vai chegar ao seu destino, andando pra frente, ou pra tras (cont++ ou cont--)

For

- Exemplo: Quero que meu programa repita 50 vezes

```
#include <stdio.h>
main(){
    int cont;
    for(cont = 0;  cont < 100; cont++){
    }
}
```

For

- Percebam que no exemplo anterior:

1. cont=0; (ponto de partida)
2. cont < 100; (nesse caso o cont pode ir no máximo até 100, ou seja, o ponto de chegada é 100)
3. cont++; (como o cont vai de 0 ate 100, ele tem que ir aumentando concordam?). Nesse caso ele vai aumentando de um em um, até chegar em 100, onde o for vai parar.

Casos de teste

- Todo maratonista que se preze já se esbarrou com um problema com algo parecido com isso:

  " A entrada contém um inteiro N que indica o número de casos de teste. Cada caso de teste.... "

- Você provavelmente deve ter pensado: "Ai meu Deus que negócio é esse de caso de teste?? Meu Deus desisto dessa vida, vou vender minha arte na praia"
- Mas calma, a questão é que as vezes um programa deve ser repetido várias vezes resolvendo determinado problema (varias vezes).
- E cada repetição dessa é chamada de caso de teste.

Casos de teste

- Exemplo: URI 1589

```
#include <stdio.h>
int main(){
    int t, a, b;
    scanf("%d", &t);
    for(int i = 0; i < t; i++){
        scanf("%d %d", &a, &b);
        printf("%d\n", a+b);
    }
}
```

EOF(End of file)

- Pode aparecer também algo assim:

" A entrada contém vários casos de teste e termina com EOF. Cada caso de teste inicia com um inteiro N. ... "

- Pra resolver isso, primeiro vamos entender como o URI ou os corretores das maratonas funcionam..
- Quando o URI vai corrigir seu programa, ele armazena todas as entradas em um arquivo, e executa seu código dizendo pra ele: "leia os valores de entrada a
- O EOF(End of File / Fim do arquivo) é um sinal que o scanf manda que indica pro seu programa que o arquivo de entrada(que o corretor est á utilizando) acabou. Ou seja, você vai ler a entrada ENQUANTO(while) o arquivo de entrada ainda não tiver terminado.

EOF(End of file)

- Exemplo:

```
#include <stdio.h>
main(){
    int n;
    while( scanf("%d", &n)  != EOF ){   //lê enquanto o arquivo
        // código
    }
}
```

E para finalizar...

- Exercicios URI
- 1059
- 1067
- 1073
- 1113
- 1144
- 1146