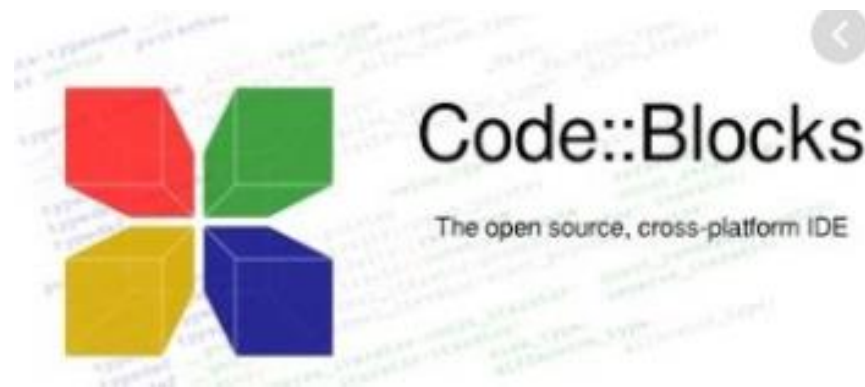


Curso Profissional de Técnico de Eletrónica, Automação e Comando | 10.º ano

AUTOMAÇÃO E COMANDO

UFCD 6052 – PROGRAMAÇÃO – INICIAÇÃO



ESTRUTURAS DE CONTROLO

Estrutura de decisão com base numa condição "IF... "

if (condição)

```
{ /* comandos aqui */  
}
```

Exemplo:

```
if (a == 3)  
{  
    printf("Os números são iguais!\n");  
}
```

O funcionamento:

- Quando a execução do programa chega ao `if`, a condição é avaliada, e há duas saídas possíveis:
 - 1- se a condição for **verdadeira**, o programa executa os comandos colocados dentro das chavetas, e depois continua a executar os comandos fora delas; •
 - 2- se a condição for **falsa**, o programa simplesmente salta o que está entre chavetas e continua a executar os comandos após as estas.

ESTRUTURAS DE CONTROLO

Estrutura de decisão com base numa condição "IF... else"

```
if (condição) {  
    /* condição for satisfeita, executa  
    estes comandos */  
}  
  
else {  
    /* SENÃO, executa estes comandos  
    */ }  
}
```

Exemplo:

```
int main()  
{  
    int num;  
    printf("Digite um número: ");  
    scanf("%d", &num);  
    if (num % 2 == 0)  
        printf("O número é par.\n");  
    else printf("O número é ímpar.\n");  
    return 0; }  
}
```

Funcionamento:

- Quando a execução do programa chega ao if, a condição é avaliada, e há duas saídas possíveis:
 - 1-se a condição for **verdadeira**, o programa executa os comandos colocados dentro das chavetas, e depois continua a executar os comandos fora delas;
 - 2- se a condição for **falsa**, o programa executa o comando else.

ESTRUTURAS DE CONTROLO

Estrutura de repetição: While...

while (condição)

```
{  
/* comandos aqui */  
}
```

Exemplo:

```
int main()  
{  
int num;  
num = 1;  
while (num <= 10) {  
printf("%d\n", num);  
num = num + 1;  
}  
return 0;  
}
```

Funcionamento:

- O bloco de comandos é executado **enquanto** a condição for verdadeira.
- Este exemplo utiliza uma variável contadora, que varia de uma unidade em cada iteração.

ESTRUTURAS DE CONTROLO

Estrutura de repetição: Do...While

do

```
{  
/* comandos aqui */  
}
```

while (condição)

Exemplo:

```
int main()  
{  
int num;  
num = 1;  
do {  
printf("%d\n", num);  
num = num + 1;  
}  
while (num <= 10)  
return 0;  
}
```

Funcionamento:

- O teste da condição neste caso só é realizado no fim.
- Este ciclo é executado pelo menos uma vez.

ESTRUTURAS DE CONTROLO

Estrutura de repetição: For...

For (inicialização; controle; incremento)

```
{  
/* comandos aqui */  
}
```

Exemplo:

```
int main()  
{  
    int num;  
    for (num = 1; num <= 10; num = num + 1)  
        printf("%d\n", num);  
    return 0;  
}
```

Funcionamento:

- A variável do contador é inicializada (num=1); isso ocorre apenas uma vez, antes do início do ciclo.
- A condição de controle especifica que condição a variável do contador deve satisfazer para que os comandos sejam executados (num<=10).
- Sempre que a condição de controle for satisfeita, é executada uma iteração do bloco de comandos fornecido.

ESTRUTURAS DE CONTROLO

Estrutur: Break...

Exemplo:

```
int main()
{
    int i, n;
    n=20;
    for (i = 2; i < n; i++)
    {
        if (n=10) break;
    }
}
```

Funcionamento:

Podemos usar a instrução **break** para terminar um ciclo em qualquer momento

- antes de uma iteração de while ou for
- depois de uma iteração de do...while

ESTRUTURAS DE CONTROLO

Exercícios:

Ex.1)

```
int main()
{
    int i;
    i = 1;
    while (i < 10) {
        printf("%d\n", i);
        i = i * 2;
    }
    Return o;
}
```

Compilador C- <https://www.programiz.com/c-programming/online-compiler/>

Resolve agora:

- 1.1- Qual é o output?
- 1.2- Substitui o ciclo while pelo ciclo For.
- 1.3- Substitui o ciclo while pelo ciclo do ..while.

2- Escreva um programa que peça ao utilizador dois números inteiros e escreva todos os números compreendidos nesse intervalo.