



paixão por educar

GRADUAÇÃO PRESENCIAL
1º semestre- 2015

Microprocessadores e
Microcontroladores

Tec. Aut. Industrial – 5º/ 6º
semestres

Profº. Ms.Cristiano Malheiro

cmalheiro@anhanguera.com

<http://cristianotm.wix.com/notasdeaula>

1



Aula 6- Seção 2.1

A Linguagem C

- “C”, ou “Linguagem C” é uma linguagem altamente adaptável (de sintaxe pequena, poderosa, transportável), desenvolvida por Dennis Ritchie nos laboratórios Bell, em 1972.
- Muitas das idéias de C vêm da linguagem mais antiga BCPL, desenvolvida por Martin Richards.



Aula 6

Um Programa Inicial

diretiva de compilação

```
#include <stdio.h>
void main(void)
{
    printf (" Universidade Bandeirante Anhanguera ");
}
```

função principal

corpo da função

3



Aula 6

Tipos de Dados Básicos

- A Linguagem C possui basicamente os seguintes tipos de dados:

- char um único byte, que pode conter um caracter do conjunto de caracteres;
- int um inteiro;
- float ponto flutuante em precisão simples;

4



Aula 6

Tipos de Dados Básicos

- Exemplos de declaração:

```
Int    a,b,c;  
char   y,nomes[100];
```

ou, realizando inicializações imediatas ...

```
int     y=0;  
float   valor=1.0e-5
```

5



Aula 6

Exemplo

O exemplo abaixo armazena na variável "c" os valores de "a" e "b":

```
#include <stdio.h>  
  
void main(void)  
{  
    int a, b, c;  
    a=10;  
    b=10;  
    c = a + b;  
    printf( "valor total: %d", c);  
}
```

6





Aula 6

A função “printf”

```
int printf(const char*format,arg_list)
```

- A função printf() escreve na saída padrão do sistema (vídeo, stdout) os argumentos descritos em arg_list, sob controle da String apontada por format.

7



Aula 6

A função “printf”

```
int printf(const char*format,arg_list)
```

- A String apontada por format pode possuir argumentos que representam caracteres que serão impressos na tela e caracteres que definem o formato pelo qual estes caracteres serão impressos.

Código	Formato
%c	Um único caracter
%d	Decimal
%%	Imprime um sinal %
%f	Ponto flutuante decimal

Código (Sequência de Escape)	Descrição
\n	Representa um avanço de linha
\t	Representa uma tabulação

8



Aula 6

A função “scanf ”

```
int scanf(const char*format,arg_list)
```

- A função scanf() lê a entrada padrão do sistema (vídeo, stdin), carregando os argumentos descritos em arg_list, sob controle da String apontada por format.

9



Aula 6

A função “scanf ”

Para receber valores, inclua o símbolo “&” (será estudado depois)

Da mesma forma são válidos:

Código	Formato
%c	Um único caracter
%d	Decimal
%%	Imprime um sinal %
%f	Ponto flutuante decimal

10



Aula 6

Exemplo

O exemplo abaixo solicita valores "a" e "b" e apresenta a soma:

```
#include <stdio.h>

void main(void)
{
    int a,b,c;
    printf("Entre com os valores a e b:\n");
    scanf("%d",&a);
    scanf("%d",&b);
    c=a+b;
    printf("\nResultado a+b ... %d\n",c);
}
```

11



Aula 6

Operadores Relacionais

<i>Operador Relacional</i>	<i>Significado</i>
>	Maior que
>=	Maior ou igual a
<	Menor que
<=	Menor ou igual a
=	Igual
!=	Não igual

12



Aula 6

Operadores Lógicos

<i>Operador Lógico</i>	<i>Significado</i>
<code>&&</code>	"E", "AND"
<code> </code>	"OU", "OR"
<code>!</code>	"NÃO", "NOT"



Aula 6

Exemplo

```
#include <stdio.h>

void main(void)
{
    int a, b, c;
    a=10;
    b=15;
    c=20;
    if (((a>5)&&(b>a))||(c!=30))
        printf("aqui!!");
}
```



Aula 6

Exercício

Se tivermos as seguintes variáveis declaradas, qual a resposta destas comparações (responder V ou F):

`a=10; b=20; c=15`

... `(a > 10) || (b > 10)`

... `(a > 10) && (b > 10)`

... `((a>10) || (b>10)) && (c<b)`

... `(a<=b) && (c<=b)`



Aula 6

Decisão

```
if (condição)
{
    comando1;
    comando2;
    ...
}
else
{
    comando3;
    comando4;
    ...
}
```

Caso a condição seja verdadeira, os comandos “comando1, comando2, ...” serão executados.
Caso contrário, os comandos “comando3”, “comando4, ...” serão executados.
O else é opcional.





Aula 6

Exemplo:

If (segundo > 59)

```
{
segundo=0;      // reseta a variável segundo
++ minuto;     // incrementa a variável minuto
}
```

Else

```
{
++segundo;     // incrementa a variável segundo
}
```

17



Aula 6

Loop (Laço)

- Existem 3 estruturas de loop em C:
- while
- do...while
- for





Aula 6

while

```
while (condição)
{
    comando1;
    comando2;
    ...
    controle;
}
```

Enquanto a condição for verdadeira, os comandos “comando1, comando2, ...” serão executados.

Notar a importância de se incluir o **controle** da condição, em alguma parte do bloco (inclusive na própria condição!).

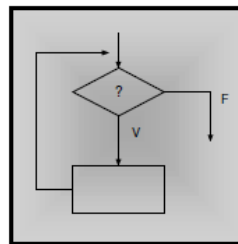
O **controle** assegura a finalização do loop.



Aula 6

while

```
while (condição)
{
    comando1;
    comando2;
    ...
    controle;
}
```





Aula 6

Exemplo

```
#include <stdio.h>
void main(void)
{
    int a;
    a=10;
    while (a<14)
    {
        printf("a=%d\n", a);
        a++;
    }
}
```

Qual a resposta
desta simulação??



Aula 6

do ... while

```
do {
    comando1;
    comando2;
    ...
    controle;
} while(condição);
```

Os comandos “comando1, comando2, ...” serão executados, enquanto a condição for verdadeira.

Notar a importância de se incluir o **controle** da condição, em alguma parte do bloco (inclusive na própria condição!).

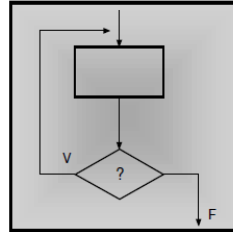
O **controle** assegura a finalização do loop.



Aula 6

do ... while

```
do {
    comando1;
    comando2;
    ...
    controle;
} while (condição);
```



Aula 6

Exemplo

```
#include <stdio.h>
void main(void)
{
    int a;
    a=10;
    do {
        printf("a=%d\n", a);
        a++;
    } while (a<14);
}
```

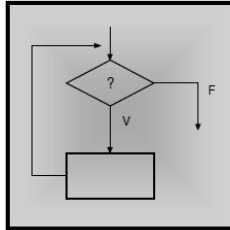
Qual a resposta
desta simulação??



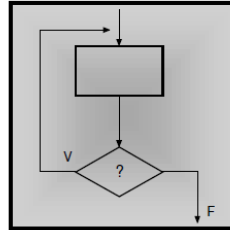
Aula 6

while X do ... while

while



do...while



Qual a diferença?



Aula 6

Os exemplos abaixo produzem o mesmo resultado, mas não são equivalentes!

```
#include <stdio.h>
void main(void)
{
    int a;
    a=10;
    while (a<14)
    {
        printf("a=%d\n", a);
        a++;
    }
}
```

```
#include <stdio.h>
void main(void)
{
    int a;
    a=10;
    do {
        printf("a=%d\n", a);
        a++;
    } while (a<14);
}
```

Porquê??



Aula 6

Os exemplos abaixo NÃO produzem o mesmo resultado, e não são equivalentes!

```
#include <stdio.h>
void main(void)
{
    int a;
    a=100;
    while (a<14)
    {
        printf("a=%d\n", a);
        a++;
    }
}
```

```
#include <stdio.h>
void main(void)
{
    int a;
    a=100;
    do {
        printf("a=%d\n", a);
        a++;
    } while (a<14);
}
```

Porquê??

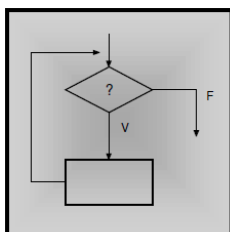


Aula 6

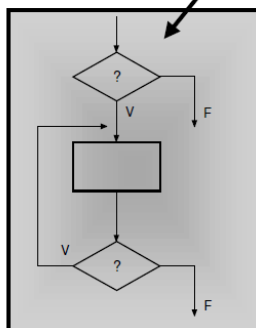
Conversão Genérica

while → do ... while

while



do...while





Aula 6

Com o ajuste, os exemplos abaixo produzem o mesmo resultado!

```
#include <stdio.h>
void main(void)
{
    int a;
    a=100;
    while (a<14)
    {
        printf("a=%d\n", a);
        a++;
    }
}
```



```
#include <stdio.h>
void main(void)
{
    int a;
    a=100;
    if (a<14)
    {
        do {
            printf("a=%d\n", a);
            a++;
        } while (a<14);
    }
}
```

Porquê??



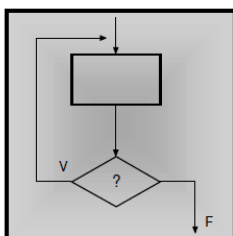
Aula 6

Conversão Genérica

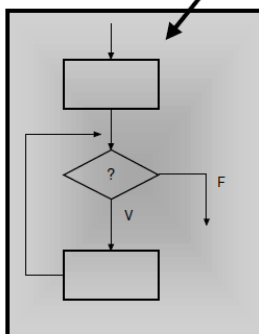
do ... while → while

Repete-se o
bloco de comando !

do...while



while





Aula 6

Com o ajuste, os exemplos abaixo produzem o mesmo resultado!

```
#include <stdio.h>
void main(void)
{
    int a;
    a=100;
    do {
        printf("a=%d\n", a);
        a++;
    } while (a<14);
}
```



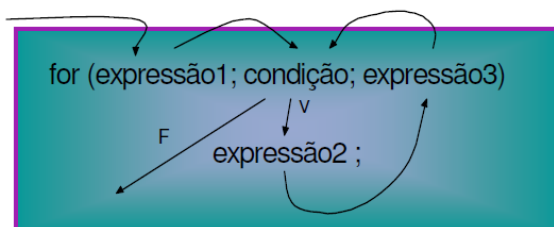
```
#include <stdio.h>
void main(void)
{
    int a;
    a=100;
    printf("a=%d\n", a);
    a++;
    while (a<14)
    {
        printf("a=%d\n", a);
        a++;
    }
}
```

Porquê??



Aula 6

for



- **expressão1** : uma expressão ou qualquer comando ou chamada de função. Normalmente uma atribuição;
- **expressão2** : qualquer comando ou chamada de função;
- **expressão3** : uma expressão ou qualquer comando ou chamada de função. Normalmente um incremento;



Aula 6

Exemplo

```
#include <stdio.h>
void main(void)
{
    int a;
    for(a=10; a<14; a++)
    {
        printf("a=%d\n", a);
    }
}
```

Qual a resposta
desta simulação??

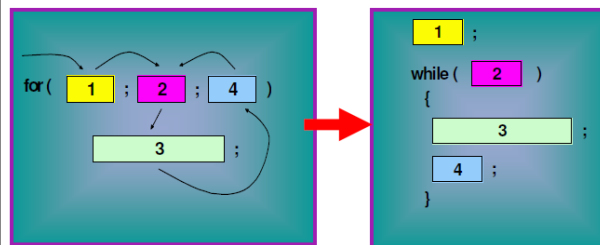
33

kroton
paixão por educar



Aula 6

O "for" é um while !



34

kroton
paixão por educar



Aula 6

Os exemplos abaixo produzem o mesmo resultado!

```
#include <stdio.h>
void main(void)
{
    int a;
    for(a=10; a<14; a++)
    {
        printf("a=%d\n", a);
    }
}
```



```
#include <stdio.h>
void main(void)
{
    int a;
    a=10;
    while(a<14)
    {
        printf("a=%d\n", a);
        a++;
    }
}
```

Porquê??



Bibliografia desta aula:

- Schildt, Herbert – C Completo e Total – Makron Books editora, 1999

