

Escrendo na tela

MC102

Algoritmos e Programação de Computadores

Aula de Laboratório 03

Instituto de Computação
Primeiro Semestre de 2012

19 de março de 2012



- Printf (*Print formatted*) é utilizado para exibir a saída do programa (imprimir texto);
- Formato: `printf("<texto>",<variável1>,...,<variávelN>);`
- Pode ser imprimir Strings;

Exemplo

```
printf("Bom dia!");
printf("Bom dia!\n Bom dia");
```

- Pode imprimir variáveis (no texto);

Exemplo

```
printf("Meu nome é:%s e tenho %d anos",nome,idade);
```



MC102 - Algoritmos e Programação de Computadores

3 / 20

Conteúdo

- 1 Escrita
- 2 Leitura
- 3 Expressões
- 4 Condicionais
- 5 Susy

Escrendo na tela

Formatos para Printf:

Código	Função
<code>%d</code>	Escreve um inteiro
<code>%u</code>	Escreve um inteiro sem sinal
<code>%ld</code>	Escreve um inteiro longo
<code>%f</code>	Escreve um ponto flutuante
<code>%.2f</code>	Escreve um float com 2 casas decimais
<code>%e</code>	Escreve usando notação científica
<code>%lf</code>	Escreve um double
<code>%c</code>	Escreve um caracter
<code>%s</code>	Escreve uma string



MC102 - Algoritmos e Programação de Computadores

2 / 20



MC102 - Algoritmos e Programação de Computadores

4 / 20

Lendo a partir do teclado

- `scanf` (*Print formatted*) é utilizado para ler dados a partir do teclado;
- Formato: `scanf ("<Tipos>",<lista de variáveis>);`

Exemplo

```

1 #include <stdio.h>
2 int main(){
3     int a,b;
4     printf("Digite dois numeros: ");
5     scanf("%d %d",&a,&b);
6     printf("Os valores foram: %d e %d \n",a,b);
7 }
```



Escrendo na tela

Formatos para `scanf`:

Código	Função
<code>%d</code>	Lê um inteiro
<code>%u</code>	Lê inteiro sem sinal
<code>%ld</code>	Lê inteiro longo
<code>%f</code>	Lê ponto flutuante
<code>%lf</code>	Lê um double
<code>%c</code>	Lê um caracter
<code>%s</code>	Lê uma string



Operadores Aritméticos

- Alguns operadores em C: `+`, `-`, `*`, `/`, `%`
- A precedência é:
 - 1 `*` e `/`;
 - 2 `%` → resto da divisão;
 - 3 `+` e `-`;
- A precedência pode ser alterada usando parênteses: (`<expressão>`);
Ex.: $4 + 3 * 2 = 10$, enquanto $(4 + 3) * 2 = 14$



Operadores Aritméticos

Operadores de incremento(`++`) e decremento(`--`)

- `a++` → incrementa em uma unidade;
- `a--` → decrementa em uma unidade;

Exemplo

```

1 #include <stdio.h>
2 int main(){
3     int a = 10;
4     printf("Valor: %d\n",a++);
5     printf("Valor: %d\n",++a);
6     printf("Valor: %d\n",a);
7 }
```



Operadores

Atribuições simplificadas:

Comando	Exemplo	Corresponde a
+=	a += b	a = a + b
-=	a -= b	a = a - b
*=	a *= b	a = a * b
/=	a /= b	a = a / b
%=	a %= b	a = a % b

Expressões lógicas

Operadores lógicos são:

Comando	Exemplo	Corresponde a
&&	E	V se ambas são verdadeiras!
 	OU	V se pelo menos uma é verdade
!	NÃO	Nega a expressão



Expressões relacionais

Formato: <expressão> <operador> <expressão> ;

Operadores relacionais são:

Comando	Exemplo	Corresponde a
==	a == b	V se iguais!
!=	a != b	V se diferentes
>	a > b	V se a expressão à esquerda é maior
<	10 < 5	V se a expressão à direita é maior
>=	10 >= 10	V se a expressão à esquerda é maior ou igual
<=	10 <= 12	V se a expressão à direita é maior ou igual



Condicionais

- If é o principal comando condicional e C;
- ```
if(<expressão lógica>) {
 comandos;
}
```

### Exemplo

```
1 #include <stdio.h>
2
3 int main(){
4 float nota = 6.0;
5 if (nota >= 5){
6 printf("Aluno aprovado!\n");
7 }
8 }
```



## Comando If

- Uma variação do If é o If-else;
 

```
if(<expressão lógica>){
 comandos;
} else {
 comandos;
}
```

### Exemplo

```

1 #include <stdio.h>
2
3 int main(){
4 float nota = 4.5;
5 if(nota >= 5){
6 printf("Aluno Aprovado!\n");
7 } else {
8 printf("Aluno Reprovado!\n");
9 }
10 }
```



## Comando Switch

- Simplifica códigos onde existem vários if/else/if's encaixados;
 

```
Switch(variável inteira){
 caso valor:
 comandos;
 break;
 caso valor:
 comandos;
 break;
}
```



## Comando If

- Outra variante do If é o If-else-if;
 

```
if(<expressão lógica>
{ ... } else if { ... }
```

### Exemplo

```

1 #include <stdio.h>
2 int main(){
3 float nota;
4 printf("Digite a nota: ");
5 scanf("%f",¬a);
6 if(nota >= 5.0){
7 printf("Aluno Aprovado!\n");
8 } else if(nota >= 4.0){
9 printf("Aluno em Exame!\n");
10 } else {printf("Aluno Reprovado!\n");}
11 }
```



## Comando Switch

### Exemplo

```

1 #include <stdio.h>
2
3 int main(){
4 int opcao;
5 printf("Digite uma opcao: [1] somar e [2] subtrair:
6 ");
7 scanf("%d",&opcao);
8 switch(opcao){
9 case 1: printf("Realizar a soma de dois valores
10 !\n");
11 break;
12 case 2: printf("Realizar a subtracao de dois
13 valores!\n");
14 break;
15 default: printf("Opcao incorreta!");
16 }
17 }
```



## Exercício

- Faça uma mini calculadora utilizando o comando Swith. A calculadora deve ler dois inteiros e em seguida capturar a opção do usuário, se é somar, subtrair, dividir ou multiplicar os dois números;



## Laboratório 00

- Já fizeram o Laboratório 00?;



## Laboratório 00

- Já fizeram o Laboratório 00?;

### Solução

```

1 /* Programa: media.c */
2 /* Autor: Fulano de Tal RA000000 */
3 /* Disciplina: MC102 Turma A */
4 /* Data: 20/mar/2012 */
5
6 #include <stdio.h>
7
8 int main() {
9 float num1, num2, num3, num4, result;
10
11 printf("Digite o primeiro numero:\n");
12 scanf("%f",&num1);
13 printf("Digite o segundo numero:\n");
14 scanf("%f",&num2);
15 printf("Digite o terceiro numero:\n");
16 scanf("%f",&num3);
17 printf("Digite o quarto numero:\n");
18 scanf("%f",&num4);
19
20 result = (num1 + num2 + num3 + num4)/4;
21 printf("Media igual a: %.2f\n",result);
22 return 0;
23 }

```



## Como testar o seu programa?

- Primeiro baixe os arquivos de entrada (ex. **arq1.in**) e os arquivos de saída (ex.: **arq1.res**);



## Como testar o seu programa?

- Primeiro baixe os arquivos de entrada (ex. **arq1.in**) e os arquivos de saída (ex.: **arq1.res**);
- Execute o seu programa redirecionando a entrada e a saída padrão utilizando os caracteres <(menor que) e > (maior que):  
**E.:./lab00 < arq1.in > arq1.out**



## Como testar o seu programa?

- Primeiro baixe os arquivos de entrada (ex. **arq1.in**) e os arquivos de saída (ex.: **arq1.res**);
- Execute o seu programa redirecionando a entrada e a saída padrão utilizando os caracteres <(menor que) e > (maior que):  
**E.:./lab00 < arq1.in > arq1.out**
- Agora basta comparar a saída do seu programa (**arq1.out**) com o arquivo de saída do laboratório (**arq1.res**):
  - **diff arq1.out arq1.res**
  - **diff -y arq1.out arq1.res**



## Questões?

Obrigado!

*Para informação:*

**Página dos Laboratórios (Tarefas):** <http://susy.ic.unicamp.br:9999/mc102ab>

**Página do Curso:** <http://www.lrc.ic.unicamp.br/~geraldoms/mc102>

**E-mail:**

[geraldoms\[at\]lrc.ic.unicamp.br](mailto:geraldoms[at]lrc.ic.unicamp.br)

[brhenrique.fischer\[at\]gmail.com](mailto:brhenrique.fischer[at]gmail.com)

