

9002 — Aula 04

Algoritmos e Programação de Computadores

Instituto de Engenharia – UFMT

Segundo Semestre de 2014

30 de setembro de 2014

Roteiro

- 1 Saída de dados
- 2 Entrada de dados
- 3 Exercício

Escrevendo na tela

- Podemos imprimir um de *texto puro* utilizando o comando `printf`. O texto pode ser uma constante do tipo cadeia de caracteres.

Exemplo

```
printf("Ola Pessoal!");  
Saída: Ola Pessoal!
```

- No meio da constante com a cadeias de caracteres pode haver comandos especiais. O símbolo especial `\n` é responsável por pular uma linha na saída.

Exemplo

```
printf("Ola Pessoal!  \n Ola Pessoal");  
Saída: Ola Pessoal!  
Ola Pessoal
```

Escrevendo na tela

- Podemos imprimir um de *texto puro* utilizando o comando `printf`. O texto pode ser uma constante do tipo cadeia de caracteres.

Exemplo

```
printf("Ola Pessoal!");
```

Saída: Ola Pessoal!

- No meio da constante com a cadeias de caracteres pode haver comandos especiais. O símbolo especial `\n` é responsável por pular uma linha na saída.

Exemplo

```
printf("Ola Pessoal! \n Ola Pessoal");
```

Saída: Ola Pessoal!

Ola Pessoal

Escrevendo na tela

- Podemos imprimir um de *texto puro* utilizando o comando `printf`. O texto pode ser uma constante do tipo cadeia de caracteres.

Exemplo

```
printf("Ola Pessoal!");
```

Saída: Ola Pessoal!

- No meio da constante com a cadeias de caracteres pode haver comandos especiais. O símbolo especial `\n` é responsável por pular uma linha na saída.

Exemplo

```
printf("Ola Pessoal! \n Ola Pessoal");
```

Saída: Ola Pessoal!

Ola Pessoal

Escrevendo na tela

- Podemos imprimir um de *texto puro* utilizando o comando `printf`. O texto pode ser uma constante do tipo cadeia de caracteres.

Exemplo

```
printf("Ola Pessoal!");
```

Saída: Ola Pessoal!

- No meio da constante com a cadeias de caracteres pode haver comandos especiais. O símbolo especial `\n` é responsável por pular uma linha na saída.

Exemplo

```
printf("Ola Pessoal!  \n Ola Pessoal");
```

Saída: Ola Pessoal!

Ola Pessoal

Escrevendo o conteúdo de uma variável na tela

- Podemos imprimir também o **conteúdo de uma variável** utilizando `printf`.
- Utilizamos símbolos especiais no texto para representar um trecho deve **substituído** por uma variável
- Passamos uma lista de variáveis ou constantes a serem substituídas, separadas por vírgula.

Exemplo

```
printf("A variável %s contém o valor %d", "a", a);  
imprime A variável a contém o valor 10
```

- Nesse caso, `%s` foi substituído por uma constante do tipo `string` enquanto `%d` foi substituído por uma variável do tipo `inteiro`.

Escrevendo o conteúdo de uma variável na tela

- Podemos imprimir também o **conteúdo de uma variável** utilizando `printf`.
- Utilizamos símbolos especiais no texto para representar um trecho deve **substituído** por uma variável
- Passamos uma lista de variáveis ou constantes a serem substituídas, separadas por vírgula.

Exemplo

```
printf("A variável %s contém o valor %d", "a", a);  
imprime A variável a contém o valor 10
```

- Nesse caso, `%s` foi substituído por uma constante do tipo `string` enquanto `%d` foi substituído por uma variável do tipo `inteiro`.

Escrevendo o conteúdo de uma variável na tela

- Podemos imprimir também o **conteúdo de uma variável** utilizando `printf`.
- Utilizamos símbolos especiais no texto para representar um trecho deve **substituído** por uma variável
- Passamos uma lista de variáveis ou constantes a serem substituídas, separadas por vírgula.

Exemplo

```
printf("A variável %s contém o valor %d", "a", a);  
imprime A variável a contém o valor 10
```

- Nesse caso, `%s` foi substituído por uma constante do tipo `string` enquanto `%d` foi substituído por uma variável do tipo `inteiro`.

Escrevendo o conteúdo de uma variável na tela

- Podemos imprimir também o **conteúdo de uma variável** utilizando `printf`.
- Utilizamos símbolos especiais no texto para representar um trecho deve **substituído** por uma variável
- Passamos uma lista de variáveis ou constantes a serem substituídas, separadas por vírgula.

Exemplo

```
printf("A variável %s contém o valor %d", "a", a);  
imprime A variável a contém o valor 10
```

- Nesse caso, `%s` foi substituído por uma constante do tipo `string` enquanto `%d` foi substituído por uma variável do tipo inteiro.

Escrevendo o conteúdo de uma variável na tela

- Podemos imprimir também o **conteúdo de uma variável** utilizando `printf`.
- Utilizamos símbolos especiais no texto para representar um trecho deve **substituído** por uma variável
- Passamos uma lista de variáveis ou constantes a serem substituídas, separadas por vírgula.

Exemplo

```
printf("A variável %s contém o valor %d", "a", a);  
imprime A variável a contém o valor 10
```

- Nesse caso, `%s` foi substituído por uma constante do tipo `string` enquanto `%d` foi substituído por uma variável do tipo inteiro.

Formatos inteiros

`%d` — Escreve um inteiro na tela.

Exemplo

```
printf ("%d", 10);  
imprime 10
```

Exemplo

```
int a=12;  
printf ("0 valor e %d", a);  
imprime 0 valor e 12
```

Formatos inteiros

- A letra **d** pode ser substituída pelas letras **u** e **l**, ou as duas, quando desejamos escrever variáveis do tipo unsigned ou long, respectivamente.

Exemplo

```
printf ("%d", 4000000000);  
escreve -294967296 na tela, enquanto que  
printf ("%u", 4000000000);  
escreve 4000000000.
```

Formatos ponto flutuante

`%f` — Escreve um ponto flutuante na tela.

Exemplo

```
printf ("%f", 10.0);  
imprime 10.000000
```

Formatos ponto flutuante

`%e` — Escreve um ponto flutuante na tela, em notação científica

Exemplo

```
printf ("%e", 10.02545);  
imprime 1.002545e+01
```

Formatos ponto flutuante

`%.< decimais >f` — Escreve um ponto flutuante na tela, com
< *decimais* > casas decimais.

Exemplo

```
printf (("%.2f", 10.1111);  
imprime 10.11
```


Formatos ponto flutuante

- A letra **f** pode ser substituída pelas letras **lf**, para escrever um double ao invés de um float

Exemplo

```
printf ("%6.2lf", 10.0);  
imprime < espaco >10.00
```

Formato caracter

`%c` — Escreve uma letra.

Exemplo

```
printf ("%c", 'A');  
imprime A
```

Note que `printf ("%c", 65)` também imprime a letra A.

Formato string

`%s` — Escreve uma string

Exemplo

```
printf ("%s", "Meu primeiro programa");  
imprime Meu primeiro programa
```

A função scanf

- Realiza a leitura de um texto a partir do teclado.
- Parâmetros:
 - ▶ Uma cadeia de caracteres, indicando os tipos das variáveis que serão lidas e o formato dessa leitura.
 - ▶ Uma lista de variáveis.
- Aguarda que o usuário digite um valor e atribui o valor digitado à variável.

A função scanf

O programa abaixo é composto de quatro passos:

- 1 Cria uma variável `n`;
- 2 Escreve na tela Digite um número:
- 3 Lê o valor do número digitado
- 4 Imprime o valor do número digitado

Exemplo

```
#include <stdio.h>
int main(){
    int n;
    printf("Digite um número: ");
    scanf("%d",&n);
    printf("O valor digitado foi %d\n",n);
    return 0;
}
```

A função scanf

Leitura de várias variáveis

```
#include <stdio.h>
main(){
    int m, n, o;
    printf("Digite três números: ");
    scanf("%d %d %d",&m, &n, &o);
    printf("0 valores digitados foram\
        %d %d %d\n", m, n, o);
    return 0;
}
```

Formatos de leitura de variável

Os formatos de leitura são muito semelhantes aos formatos de escrita utilizados pelo `printf`. A tabela a seguir mostra alguns formatos possíveis de leitura

Código	Função
<code>%c</code>	Lê um único caracter
<code>%s</code>	Lê uma série de caracteres
<code>%d</code>	Lê um número decimal
<code>%u</code>	Lê um decimal sem sinal
<code>%l</code>	Lê um inteiro longo
<code>%f</code>	Lê um número em ponto flutuante
<code>%lf</code>	Lê um double

Um parêntese: comentários

- O código fonte pode conter comentários direcionados unicamente ao programador. Estes comentários devem estar delimitados pelos símbolos `/*` e `*/`, e são ignorados pelo compilador.

Exemplo

```
#include <stdio.h>

/* Este é o meu primeiro programa. */
//Isto também é um comentário
main() {
    printf("Hello, world!\n");
    return 0;
}
```

- Comentários são úteis para descrever o algoritmo usado e para explicitar suposições não óbvias sobre a implementação.

Exercício

Resolução da segunda lista de exercícios.

Nas próximas aulas...

- Veremos como realizar decisões em nossos algoritmos :)
- FIM!!