

Curso de C para Engenharías

Aula 8

Cristiano Dalbem Dennis Balreira Gabriel Moreira
Miller Biazus Raphael Lupchinski

Universidade Federal do Rio Grande do Sul
Instituto de Informática
Grupo PET Computação

Bibliotecas

math.h

Funções matemáticas

- Trigonométricas
- Hiperbólicas
- Logarítmicas e exponenciais
- Potência
- Arredondamento, resto e valor absoluto

Algumas funções

- pow (**double** x, **double** n)
 - x elevado na *n*
- sqrt (**double** a)
 - Calcula a raiz quadrada de *a*
- sin (**double** grau)
 - Retorna o seno de *grau* (deve estar em radianos)
- cos(**double** grau)
 - Retorna o cosseno de *grau* (deve estar em radianos)
- fabs(**double** num)
 - Retorna valor absoluto de *num*

Exemplo

```
1  #include <stdio.h>
2  #include <math.h>
3
4  int main ()
5  {
6      double x, y;
7      printf("Digite dois numeros: ");
8      scanf("%lf %lf", &x, &y);
9      printf("x^y = %lf", pow(x,y) );
10     printf("x^(1/2) = %lf", sqrt(x) );
11     printf("cos(y) = %lf", cos(y) );
12     printf("valor absoluto de x = %lf", fabs(y) );
13     return 0;
14 }
```

conio2.h

Modo gráfico em linha de comando

- Permite manipular componentes gráficas pela interface de linha de comando.

Constantes para cores

```
1 enum    COLORS {  
2     BLACK, BLUE, GREEN, CYAN,  
3     RED, MAGENTA, BROWN, LIGHTGRAY,  
4     DARKGRAY, LIGHTBLUE, LIGHTGREEN, LIGHTCYAN,  
5     LIGHTRED, LIGHTMAGENTA, YELLOW, WHITE  
6 }
```

Função para mudar a coordenada

```
1 gotoxy(x,y)
2 // Modifica a posição do cursor para a coordenada especificada por x e y

1 // Exemplo:
2 #include <conio2.h>
3 int main() {
4     gotoxy(10,10);
5     printf("P -> P eh o ponto (10,10)");
6     return(0);
7 }
```


Funções para limpar

```
1 clrscr()  
2 // Limpa a tela  
  
1 clreol()  
2 // Limpa o resto da linha a partir do cursor  
  
1 // Exemplo:  
2 #include <conio2.h>  
3 int main() {  
4     gotoxy(30,14);  
5     printf("SUJANDO A TELA");  
6     gotoxy(30,14);  
7     clreol();  
8     printf("DE NOVO");  
9     clrscr();  
10    return(0);  
11 }
```

Funções para colorir

```
1 textbackground(cor)
2 // Altera a cor de fundo - usar constantes especificadas para escolher a
  cor

1 textcolor (cor)
2 // Altera a cor do texto - usar constantes especificadas para escolher a
  cor

1 // Exemplo:
2 #include <conio2.h>
3 int main() {
4     textbackground(LIGHTBLUE);
5     clrscr();
6     textcolor(RED);
7     printf("FUNDO EM AZUL CLARO E TEXTO EM VERMELHO");
8     return(0);
9 }
```

Funções para escrever

```
1 cputsxy(x, y, string)
2 // Coloca a string desejada nas coordenadas especificadas por x e y
3 putchxy(x, y, caractere)
4 // Coloca o caractere desejado na posição especificada por x e y
```

```
1 // Exemplo:
2 #include <conio2.h>
3 int main() {
4     cputsxy(11,5,"estilo printf sem gotoxy");
5     putchxy(40,20,'C');
6     putchxy(40,21,'h');
7     putchxy(40,22,'a');
8     putchxy(40,23,'r');
9     return(0);
10 }
```

Outras funções

```
1 getch ()
2 // Espera pressionar botão e retorna o caractere pressionado
3 wherex ()
4 // Retorna o lugar da posição x
5 wherey ()
6 // Retorna o lugar da posição y
7 delline ()
8 // Deleta a linha atual, puxando todas as linhas abaixo
9 insline ()
10 // Insere uma linha em branco a partir da linha atual
```

Exemplo

```
1  #include <stdio.h>
2  #include <conio2.h>
3  int main (){
4      int x,y;
5      char ch;
6      gotoxy(x,y);
7      textcolor(WHITE);
8      do{
9          ch = getch();
10         if(ch == 27)
11             printf("ENTER PRESSIONADO");
12         if(ch == 'a') printf("a");
13     }
14     while(ch != 'q');
15     return(0);
16 }
```

Outras bibliotecas

- `stdlib.h` - biblioteca padrão, com vários tipos e conversão.
- `time.h` - define funções de manipulação de tempo.
- `ctype.h` - funções de classificação e mapeamento de conjuntos.
- `windows.h` - manipulação de janelas e arquivos no sistema operacional windows.
- `file.h` - funções extras para manipulação de arquivos.
- OpenGL, GTK, SDL - especialização que define funções gráficas 3D e 2D. Utilizado para aplicações gráficas.

Pré-Processador

- Consiste em um programa que roda sobre o nosso código fonte antes que ocorra a compilação.
- Inclui o código de um arquivo em outro (diretiva **#include**)
- Substitui os valores no código (diretiva **#define**)

Inclusão de arquivos

- **#include** <arquivo.h> - arquivo de um diretório padrão
- **#include** "arquivo.h" - arquivo dentro da pasta atual

Substituição de #defines

- Já foi falado diversas vezes
- **#define** NOME valor

Cuidado!

- A substituição se dá a nível textual
- Isso pode causar alguns problemas...

NÃO colocar o ; no final

```
1 //um erro classico
2 #define MAX 10;
3
4 int main()
5 {
6     int array[MAX]; // int array[10;];
7     ... //seu codigo
8     return 0;
9 }
```

Cuidado com as expressões

```
1 #define SEIS 1+5
2 #define NOVE 8+1
3
4 int main()
5 {
6     printf("6 * 9 = %d". SEIS*NOVE); // 6 * 9 = 42, ahnn??
7     return 0;
8 }
```

Headers

- São os arquivos .h
- Úteis para incluir protótipos de funções, e definições de estruturas

Arquivo exemplo

```
1 //arquivo Ponto.h
2 struct Ponto {
3     int x;
4     int y;
5     int z;
6 };
7
8 typedef struct Ponto Ponto;
```

Exercícios

Exercício

Jogo da velha

- Desenvolva um jogo da velha com interface gráfica utilizando a biblioteca conio2.h.

Revisão

Aula 1

- Tipos de dados - int, float, char, double, void
- printf - Escrever na linha de comando.
- scanf - Ler variável da linha de comando.
- Operadores Aritméticos - +, -, *, /, %, =.
- Algoritmos.

Aula 2

- Fluxogramas.
- Comandos condicionais - if-else, switch.
- Operadores relacionais - >, >=, <, <=, ==, !=.
- Operadores booleanos - !, &&, ||.

Aula 3

- Comandos iterativos - for, while, do-while.
- Vetores e matrizes.

Aula 4

- Strings - strcmp, strcpy, strlen, strcat.
- Ponteiros.

Aula 5

- Funções.
- Passagens de parâmetros - valor e referência.
- Recursão.

Aula 6

- Estruturas.
- Enumerações.

Aula 7

- Arquivos - texto e binário.

Aula 8

- Bibliotecas - `math.h`, `conio.h`.
- Pré-processamento - `include`, `define`.

Dúvidas, Sugestões, Críticas

www.inf.ufrgs.br/pet
pet@inf.ufrgs.br