DeVry Brasil | UNIFAVIP

Linguagem de Programação C
Aula 5

Professor: Rodrigo Rocha

Conteúdo - Entradas e Saídas

- printf
- scanf
- getch
- getche
- gets
- puts
- fflush

Conteúdo - Entradas e Saídas

- A função **printf()** tem a seguinte forma geral:

```
printf (string_de_controle, lista_de_argumentos);
```

- Teremos, na string de controle, uma descrição de tudo que a função vai colocar na tela. A string de controle mostra não apenas os caracteres que devem ser colocados na tela, mas também quais as variáveis e suas respectivas posições.

Código	Significado
%d	Inteiro
%f	Float
%c	Caractere
%s	String
%%	Coloca na tela um %

Conteúdo - EntradasFunção: printf e Saídas

- String de Controle - Formato:

% *[flag] [tamanho] [.precisão] tipo*

[.precisão]

especificador de precisão, dígitos a direita do ponto decimal. (Opcional)

`(nada)` padrão: 6 dígitos para reais.

`.0` nenhum digito decimal.

FUNÇÃO: PRINTF

Uso dos modificadores de formato :

int i = 12, j = -35, k = 9386;
float r = 5.83, s = -82.3, t = 5467.75;

```
printf("\n\nJustificacao a direita");
printf("\n %6d  %12f",i,r);
printf("\n %6d  %12f",j,s);
printf("\n %6d  %12f",k,t);

printf("\n\nJustificacao a esquerda");
printf("\n %-6d  %-12f",i,r);
printf("\n %-6d  %-12f",j,s);
printf("\n %-6d  %-12f",k,t);
```

Constantes de barra invertida

- O C utiliza, para nos facilitar a tarefa de programar, vários códigos chamados códigos de barra invertida.

Código	Significado
\b	Retrocesso ("back")
\f	Alimentação de formulário ("form feed")
\n	Nova linha ("new line")
\t	Tabulação horizontal ("tab")
\"	Aspas
\'	Apóstrofo
\0	Nulo (0 em decimal)
\\	Barra invertida
\v	Tabulação vertical
\a	Sinal sonoro ("beep")
\N	Constante octal (N é o valor da constante)
\xN	Constante hexadecimal (N é o valor da constante)

Função: printf - Exemplo

- printf("Teste %% %%")
  - Escreve: "Teste % %"
- printf("%f",40.345)
  - Escreve: "40.345"
- printf("Um caractere %c e um inteiro %d",'D',120)
  - Escreve: "Um caractere D e um inteiro 120"
- printf("%s e um exemplo","Este")
  - Escreve: "Este e um exemplo"
- printf("%s%d%%","Juros de ",10)
  - Escreve: "Juros de 10%"

Função: scanf

- O formato geral da função **scanf()** é:

```
scanf (string-de-controle,lista-de-
  argumentos);
```

- Usando a função **scanf()** podemos pedir dados ao usuário.

- Mais uma vez, devemos ficar atentos a fim de colocar o mesmo número de argumentos que o de códigos de controle na string de controle.

Tipos de Dados em C

Tipo	Num de bits	Formato para leitura com scanf	Intervalo	
			Inicio	Fim
char	8	%c	-128	127
unsigned char	8	%c	0	255
signed char	8	%c	-128	127
int	16	%i	-32.768	32.767
unsigned int	16	%u	0	65.535
signed int	16	%i	-32.768	32.767
short int	16	%hi	-32.768	32.767
unsigned short int	16	%hu	0	65.535
signed short int	16	%hi	-32.768	32.767
long int	32	%li	-2.147.483.648	2.147.483.647
signed long int	32	%li	-2.147.483.648	2.147.483.647
unsigned long int	32	%lu	0	4.294.967.295
float	32	%f	3,4E-38	3.4E+38
double	64	%lf	1,7E-308	1,7E+308
long double	80	%Lf	3,4E-4932	3,4E+4932

Função: scanf - Exemplo

- scanf(“%i“, &i)
  - Lê um inteiro
- scanf("%i%i%i",&i,&j,&k);
  - Lê três inteiros
- scanf(“%c”, &c);
  - Lê um caracter
- scanf("%”, s);
  - Lê uma string

Função: scanf - Exemplo

```
char l1, l2;
    printf("Insira 1c: ");
    scanf("%c",&l1);

    printf("Insira c2: ");
    scanf("%c",&l2);

    printf("Você digitou: '%c' e '%c'", l1,l2);
```

O que ocorre com esse código?

Função: scanf - Exemplo

```
char l1, l2;
    printf("Insira 1c: ");
    scanf("%c",&l1);

    printf("Insira c2: ");
    scanf("  %c",&l2);

    printf("Você digitou: '%c' e '%c'", l1,l2);
```

- Esse espaço antes da string de controle é um comando para o C desconsiderar o enter, tab ou espaço em branco.

Limpando buffer

- A função scanf, lê um dado do buffer de teclado (STDIN), mas deixa o caracter de controle, ENTER, dentro do mesmo. Isto pode ocasionar problemas, pois na próxima leitura de um caracter (getch) ou de um gets, este caracter indicará que a pessoa já digitou algo

- Para limpar o buffer em Windows, use: **fflush(stdin)**
- Para limpar o buffer em Linux, use: **__fpurge(stdin)**

Limpando buffer - Exemplo

- **char** l1, l2;
- **printf**("Insira um caractere: ");
- **scanf**("%c", &l1);
- **fflush**(**stdin**);
- //__fpurge(**stdin**);
- **printf**("Insira outro caractere: ");
- **scanf**("%c", &l2);
- **printf**("Você digitou: '%c' e '%c'", l1, l2);

Ler caracter

- Funções para leitura de um caracter da tela, sem esperar o enter

- Formato geral:

```
variável_char = getch();  // Sem eco
variável_char = getche(); // Com eco
```

- Exemplo:

```
char a, b;
a=getch();  // Lê um caracter sem mostrá-lo
            // no monitor
b=getche(); // Lê e mostra o caracter
```

Ler de string

- Função para leitura de uma string, sequencia de caracteres, da tela

- Formato geral

```
gets(variável_string);
```

- Exemplo:

```
char nome[30];
gets(nome);
```

Mostrar string

- Função para mostrar uma string, sequencia de caracteres, na tela

- Formato geral

```
puts(variável_ou_constante_string);
```

- Exemplo:

```
char nome[30]="JOSE";
puts(nome);
Puts("TESTE");
```

Manipulando String - Função strcpy()

Sintaxe:

strcpy(destino,origem);

Copia o conteúdo de uma string.

```
main(){
   char str[80];
   strcpy(str,"alo");
   puts(str);
}
```

Manipulando String - Função strcat()

Sintaxe:

```
strcat(string1,string2);
```

Concatena duas strings. Não verifica tamanho.

```
main()
{
  char um[20],dois[10];

  strcpy(um,"bom");
  strcpy(dois," dia");
  strcat(um,dois);
  printf("%s\n",um);
}
```

Manipulando String - Função strcmp()

Sintaxe:

```
strcmp(s1,s2);
```

Compara duas strings, se forem iguais devolve 0.

```
main()
{
  char s[80];
  printf("Digite a senha:");
  gets(s);
  if (strcmp(s,"laranja"))
     printf("senha inválida\n");
  else
     printf("senha ok!\n") ;
}
```

Função strcmp() – (Continuação)

- **int strcmp(str1, str2)**

Compara duas strings

Condição	Retorno
<0	Se str1 é menor que str2
0	Se str1 é igual à str2
>0	Se str1 é maior que str2

A ordem lexicográfica é utilizada para a comparação

Manipulando String - **Função strlen()**

Sintaxe:

strlen(string);

Exibe o tamanho da string.

```
Int main(){
   char str[80];
   strcpy(str,"alo");
   puts(str);
   printf("%i",strlen(str));
}
```