

Algoritmos - TADS

Algoritmos – Introdução à linguagem C

Professor: Victor Hugo L Lopes

Agenda

- Entrada e saída de dados em um programa em linguagem C;
- Exercícios propostos.

Programação em Linguagem C

- A função printf()

A instrução

Printf("primeiro programa");

É a chamada da função printf(), que dará saída no monitor da cadeia de caracteres "primeiro programa".

É uma função com argumento textual.

Programação em Linguagem C

- A função printf()

Além do ENTER, diversos outros caracteres não podem ser inseridos em uma cadeia para ser apresentada pela função printf(). Para contornar isso, faz-se uso de caracteres de escape:

<code>\n</code>	-	nova linha
<code>\t</code>	-	tabulação
<code>\\</code>	-	barra invertida
<code>\0</code>	-	numeral zero
<code>\'</code>	-	aspas simples
<code>\"</code>	-	aspas duplas

Programação em Linguagem C

- A função printf()

Para a correta saída formatada dos dados, se faz necessário a utilização de modificadores:

%c-	Caractere simples
%d	- Inteiro decimal com sinal
%u	- Inteiro decimal sem sinal
%f -	decimal
%s-	Cadeia de caracteres
%o	- Inteiro octal sem sinal
%x-	Inteiro hexadecimal sem sinal
%X	- Inteiro hexadecimal sem sinal

Coloca-se o l antes da letra pra indicar que é long

Programação em Linguagem C

- A função printf()

Ex.:

```
#include <stdio.h>//para usar printf
```

```
#include <stdlib.h>//para usar system
```

```
Int main(){
```

```
    Printf("Este é o número %d. \n", 5);
```

```
    Printf("%s esta a %d milhões de milhas\n do  
sol.\n", "Venus", 67);
```

```
    System("PAUSE");
```

```
    Return 0;
```

```
}
```

Programação em Linguagem C

- **Fixação:**

Desenvolva um programa em linguagem C para escrever uma tabela de produtos e preços, bem como suas descrições.

Liste 10 produtos. Utilize variáveis para os preços dos produtos.

Programação em Linguagem C

- Explorando a função printf():

A função printf() usa uma frase (cadeia de caracteres, ou simplesmente string) para escrever na tela um resultado ou texto desejado.

```
printf("Saida do programa");
```

Para imprimirmos um valor constante ou o valor de uma variável, utiliza-se o simbolo % seguido de uma letra, que será o formatador de saída.

```
printf("O resultado é:%d", 10);
```


Programação em Linguagem C

- Explorando a função printf():

A função printf() permite estabelecer o tamanho mínimo de um campo para impressão.

Isso significa que poderemos definir o nº de colunas que serão ocupadas por um valor a ser impresso. Muito útil em controle de formulários e tabelas:

```
/* Tamanho de campo com inteiros */  
#include <stdio.h> /* Para printf() */  
#include <stdlib.h> /* Para system() */  
int main()  
{  
    printf("Os alunos sao %2d.\n",350);  
    printf("Os alunos sao %4d.\n",350);  
    printf("Os alunos sao %5d.\n",350);  
    system("PAUSE");  
    return 0;  
}
```

Programação em Linguagem C

- Explorando a função printf():

Ex.:

```
int main(int argc, char** argv) {  
    int lapis=45, borrachas=2345, canetas=420, cadernos=8,fitas=13050;  
  
    printf("\nLapis          %12d",lapis);  
    printf("\nBorrachas        %12d",borrachas);  
    printf("\nCanetas          %12d",canetas);  
    printf("\nCadernos          %12d",cadernos);  
    printf("\nFitas            %12d",fitas);  
    system("PAUSE");  
    Return 0;  
}
```

Programação em Linguagem C

- Explorando a função printf():

Pode-se obter precisão de arredondamento de ponto flutuante.

```
int main(int argc, char** argv) {  
    float lapis=4.785, borrachas=234.542, canetas=42.036, cadernos=8.0,  
        fitas=13.050;  
  
    printf("\nLapis          %12.2f",lapis);  
    printf("\nBorrachas       %12.2f",borrachas);  
    printf("\nCanetas          %12.2f",canetas);  
    printf("\nCadernos          %12.2f",cadernos);  
    printf("\nFitas            %12.2f",fitas);  
    system("PAUSE");  
    Return 0;  
}
```

Programação em Linguagem C

- Explorando a função printf():

Alinhando à esquerda ou à direita:

```
int main(int argc, char** argv) {
```

```
    //flutuantes com alinhamento a direita
```

```
    printf("%10.2f %10.2f %10.2f\n", 8.0,15.3,584.13);
```

```
    printf("%10.2f %10.2f %10.2f\n", 834.0,1500.55,4890.12);
```

```
    //flutuantes com alinhamento a esquerda
```

```
    printf("\n\n");
```

```
    printf("%-10.2f %-10.2f %-10.2f\n", 8.0,15.3,584.13);
```

```
    printf("%-10.2f %-10.2f %-10.2f\n", 834.0,1500.55,4890.12);
```

```
    system("PAUSE");
```

```
    Return 0;
```

```
}
```

Programação em Linguagem C

- Explorando a função printf():

Completando com zeros à esquerda:

```
int main(int argc, char** argv) {
```

```
    //completando com zeros a esquerda
```

```
    printf("\n\n");
```

```
    printf("\n%04d",21);
```

```
    printf("\n%06d",21);
```

```
    printf("\n%6.4d",21);
```

```
    printf("\n%6.0d",21);
```

```
    system("PAUSE");
```

```
    Return 0;
```

```
}
```

Programação em Linguagem C

- Explorando a função printf():

Escolhendo a base numérica de saída da informação:

```
int main(int argc, char** argv) {
```

```
    //alterando a base numerica da saida
```

```
    printf("\n\n");
```

```
    printf("\n%d",65);//decimal
```

```
    printf("\n%x",65);//hexadecimal
```

```
    printf("\n%o",65);//octal
```

```
    printf("\n%c",65);//character
```

```
    system("PAUSE");
```

```
    Return 0;
```

```
}
```

Programação em Linguagem C

- Explorando a função printf():

Tamanho do campo em saídas textuais:

```
int main(int argc, char** argv) {
```

```
    //alterando a base numerica da saida
```

```
    printf("\n\n");
```

```
    printf("\n%d",65);//decimal
```

```
    printf("\n%x",65);//hexadecimal
```

```
    printf("\n%o",65);//octal
```

```
    printf("\n%c",65);//character
```

```
    system("PAUSE");
```

```
    Return 0;
```

```
}
```

Programação em Linguagem C

- Explorando a função printf():

Imprimindo caracteres gráficos (ASCII-128 a 255):

```
int main(int argc, char** argv) {
```

```
    //caracteres graficos
```

```
    //system("cls");
```

```
    printf("\n\n");
```

```
    printf("\n\t\xDC\xDC\xDB\xDB\xDB\xDB\xDC\xDC");
```

```
    printf("\n\t\xDFO\xDF\xDF\xDF\xDF0\xDF");
```

```
    system("PAUSE");
```

```
    Return 0;
```

```
}
```


Programação em Linguagem C

- Explorando a função printf():

Imprimindo caracteres gráficos (ASCII-128 a 255):

```
int main()
{
    system("cls"); /*Limpa a tela*/
    printf("\n\n");
    printf("\n\t\xC9\xCD\xBB");
    printf("\n\t\xBA \xBA");
    printf("\n\t\xC8\xCD\xBC");
    printf("\n\n");
    system("PAUSE");
    return 0;
}
```

Programação em Linguagem C

- Lendo dados de entrada do teclado???

Programação em Linguagem C

- Explorando a função scanf():

A função scanf() é outra função I/O presente na biblioteca padrão da linguagem C, fornecida com os compiladores C. Está definida em **stdio.h**.

Ela é complemento da função printf() e nos permite ler dados da entrada padrão (teclado).

Sintaxe:

scanf(“expressão de controle”, lista de argumentos);

A expressão de controle deve conter formatadores, e a lista de argumentos deve conter a(s) variável(eis) onde serão guardados os valores entrados.

Códigos de formatação para scanf()	Significado
%c	Caractere simples.
%d	Inteiro decimal com sinal.
%i	Inteiro decimal, hexadecimal ou octal.
%e	Notação científica.
%f	Ponto flutuante em decimal.
%g	Usa %e ou %f, o que for menor.
%o	Inteiro octal.
%s	String de caracteres.
%u	Inteiro decimal sem sinal.
%x	Inteiro hexadecimal.
%ld	Inteiro decimal longo.
%lf	Ponto flutuante longo (double).
%Lf	Double longo.

Programação em Linguagem C

- Explorando a função scanf():

ex.:

```
int main(int argc, char** argv) {  
  
    int anos, dias;  
    printf("\nDigite sua idade em anos: ");  
    scanf("%d",&anos);  
    dias = anos * 365;  
    printf("\nSua idade em dias e:%d",dias);  
    system("PAUSE");  
    Return 0;  
}
```

Programação em Linguagem C

- Explorando a função scanf():

Múltiplas entradas com scanf().:

```
int main(int argc, char** argv) {
```

```
    float p1, p2, p3, p4;
```

```
    double media;
```

```
    printf("\nDigite as notas das 4 provas: ");
```

```
    scanf("%f%f%f%f",&p1, &p2, &p3, &p4);
```

```
    media = (p1+p2+p3+p4)/4.0;
```

```
    printf("\nMedia: %4.2f\n",media);
```

```
    system("PAUSE");
```

```
    Return 0;
```

```
}
```

Programação em Linguagem C

- Explorando a função `scanf()`:

Fixação:

Faça um programa em C, para ler a carga horária de um curso e a quantidade de faltas de um aluno, considerando em horas.

Calcule e apresente a porcentagem de faltas que este aluno possui.

Programação em Linguagem C

- Explorando a função `getchar()`:

A função `getchar()` é uma alternativa ao `scanf`, na sintaxe utilizada abaixo:

```
char ch;  
printf("\nDigite uma tecla");  
ch = getchar(); //aguarda uma tecla do teclado  
printf("\nA tecla digitada ASCII: %c.\n", ch);
```

Presente em `stdio.h`

Programação em Linguagem C

- Explorando a função `getche()` e `getch()`:

A função `getche()` retorna o caractere digitado, sem a necessidade de aguardar que se tecle o enter.

```
char ch;  
printf("\nDigite uma tecla");  
ch = getche(); //aguarda uma tecla do teclado  
printf("\nA tecla digitada ASCII: %c.\n", ch);
```

Presente em `conio.h`

Programação em Linguagem C

- Explorando a função putchar():

A função getche() retorna o caractere digitado, sem a necessidade de aguardar que se tecle o enter.

```
char ch;  
printf("\nDigite uma tecla");  
ch = getche(); //aguarda uma tecla do teclado  
printf("\nA tecla digitada ASCII: %c.\n");  
putchar(ch);  
    //putchar('c');
```

Programação em Linguagem C

- Explorando incremento e decremento:

O uso do incremento (++) opera sobre o valor de uma variável inteira e adiciona 1 ao seu valor. Pode operar de forma préfixada ou pós-fixada.

```
Int n = 5, x;  
x = ++n;  
printf("\nN:%d – X:%d",n,x);  
// a saída será N:6 – X:6
```

```
Int n = 5, x;  
x = n++;  
printf("\nN:%d – X:%d",n,x);  
// a saída será N:6 – X:5
```

Programação em Linguagem C

- Operadores aritméticos de atribuição:

São operadores que combinam os operadores aritméticos com um operador de atribuição: += -= *= /=

<code>i +=2;</code>	equivale a	<code>i = i + 2;</code>
<code>i *=y+1;</code>	equivale a	<code>i = i * (y + 1);</code>
<code>i /=2;</code>	equivale a	<code>i = i / 2;</code>
<code>i -=2;</code>	equivale a	<code>i = i - 2;</code>

Programação em Linguagem C

- E o tipo de dados lógico?

Em C não há um tipo de dados booleano, que aceita verdadeiro(true) ou falso (false). Para isso, assume-se que falso vale 0 e verdadeiro vale 1.

<code>i +=2;</code>	equivale a	<code>i = i + 2;</code>
<code>i *=y+1;</code>	equivale a	<code>i = i * (y + 1);</code>
<code>i /=2;</code>	equivale a	<code>i = i / 2;</code>
<code>i -=2;</code>	equivale a	<code>i = i - 2;</code>