MTP – Métodos e Técnicas de Programação

Universidade Federal de Uberlândia
Prof. Renato Carrijo
renato.eletrica.ufu@gmail.com

Uso do printf

Aplicações:

- printf - *"**print f**ormatted"* (imprimir formatado).

Uso do printf

Especificações de formato:

Código	Formato
%d	Número decimal inteiro (int). Também pode ser usado %i como equivalente a %d.
%u	Número decimal natural (unsigned int), ou seja, sem sinal.
%o	Número inteiro representado na base octal. Exemplo: 3737 (corresponde ao decimal 2015).
%x	Número inteiro representado na base hexadecimal. Exemplo: 7df (corresponde ao decimal 2015). Se usarmos %X, as letras serão maiúsculas: 7DF.
%X	Hexadecimal com letras maiúsculas
%f	Número decimal de ponto flutuante. No caso da função printf, devido às conversões implícitas da linguagem C, serve tanto para **float** como para **double**. No caso da função scanf, %f serve para **float** e %lf serve para **double**.
%e	Número em notação científica, por exemplo 4.91e-11. Podemos usar %E para exibir o E maiúsculo (4.91E-11).

Uso do printf

Especificações de formato:

Código	Formato
%g	Escolhe automaticamente o mais apropriado entre %f e %e. Pode-se usar %G para escolher entre %f e %E.
%p	Ponteiro: exibe o endereço de memória do ponteiro em notação hexadecimal.
%c	Caractere: imprime o caractere que tem o código ASCII correspondente ao valor dado.
%s	Sequência de caracteres (string).
%%	Imprime um %

Uso do printf

Formatação do printf

Os formatos seguem o seguinte padrão:

%[**opções**][**largura do campo**][.**precisão**][**tamanho da variável**] **tipo de dado**

Obs.: Os itens entre [ ] são opcionais.

Opções

%[**opções**][**largura do campo**][.**precisão**][**tamanho da variável**] **tipo de dado**

0: o tamanho do campo deve ser preenchido com zeros à esquerda quando necessário, se o parâmetro correspondente for numérico.

- (hífen): o valor resultante deve ser alinhado à esquerda dentro do campo (o padrão é alinhar à direita).

(espaço): no caso de formatos que admitem sinal negativo e positivo, deixa um espaço em branco à esquerda de números positivos.

+: o sinal do número será sempre mostrado, mesmo que seja positivo.

Largura do campo

%[opções][largura do campo][.precisão][tamanho da variável] tipo de dado

Especifica a largura mínima do campo.

Se o valor não ocupar toda a largura do campo, este será preenchido com espaços ou zeros.

Pode-se imprimir, por exemplo, um código de até 5 dígitos preenchido com zeros, de maneira que os valores 3, 26, 508 e 65535 apareçam como 00003, 00026, 00508 e 65535.

Exemplo:

```
printf ("%5d", 18);   // imprime "   18"
printf ("%05d", 18);  // imprime "00018"
printf ("%-5d", 18);  // imprime "18   "
```

Precisão

%[opções][largura do campo][.precisão][tamanho da variável] tipo de dado

A precisão pode ter seguintes significados:

1) Se a conversão solicitada for inteira (**d**, **i**, **o**, **u**, **x**, **X**): o número mínimo de dígitos a exibir (será preenchido com zeros se necessário).
2) Se a conversão for real (**a**, **A**, **e**, **E**, **f**, **F**): o número de casas decimais a exibir. O valor será arredondado se a precisão especificada no formato for menor que a do argumento.
3) Se a conversão for em notação científica (**g**, **G**): o número de algarismos significativos. O valor será arredondado se o número de algarismos significativos pedido for maior que o do argumento.
4) Se a conversão for de uma sequência de caracteres (**s**): o número máximo de caracteres a exibir.

Precisão

%[opções][largura do campo][.precisão][tamanho da variável] tipo de dado

Exemplo:

```
printf ("%.5d", 219);              // imprime "00209"
printf ("%.5f", 2.4);              // imprime "2.40000"
printf ("%.5g", 23456789012345);   // imprime "2.3457e+13"
printf ("%.5s", "Bom dia");        // imprime "Bom d"
```

Uso de type cast

Aplicações:

Typecasting é uma técnica utilizada em C para simples conversão de tipo de dado.

Uso de type cast

Exemplo:

```c
#include "stdio.h"
#include "stdlib.h"

void main()
{

    int a, b;
    float c;
    a = 1;
    b = 2;
    c =  a / b;

    printf("%f", c);

    system("PAUSE");
}
```

Uso de type cast

Exemplo:

```c
#include "stdio.h"
#include "stdlib.h"

void main()
{

    int a, b;
    float c;
    a = 1;
    b = 2;
    c =  a / b;

    printf("%f", c);

    system("PAUSE");
}
```

Incorreto!!

Uso de type cast

Exemplo:

```c
#include "stdio.h"
#include "stdlib.h"

void main()
{

    int a, b;
    float c;
    a = 1;
    b = 2;
    c =  a / b;

    printf("%f", c);

    system("PAUSE");
}
```

Incorreto!!

```c
#include "stdio.h"
#include "stdlib.h"

void main()
{

    int a, b;
    float c;
    a = 1;
    b = 2;
    c = (float) a / b;

    printf("%f", c);

    system("PAUSE");
}
```

Uso de type cast

Exemplo:

```c
#include "stdio.h"
#include "stdlib.h"

void main()
{

    int a, b;
    float c;
    a = 1;
    b = 2;
    c =  a / b;

    printf("%f", c);

    system("PAUSE");
}
```

Incorreto!!

```c
#include "stdio.h"
#include "stdlib.h"

void main()
{

    int a, b;
    float c;
    a = 1;
    b = 2;
    c = (float) a / b;

    printf("%f", c);

    system("PAUSE");
}
```

Uso de type cast

Exemplo:

```c
#include "stdio.h"
#include "stdlib.h"

void main()
{

    int a, b;
    float c;
    a = 1;
    b = 2;
    c =  a / b;

    printf("%f", c);

    system("PAUSE");
}
```

Incorreto!!

```c
#include "stdio.h"
#include "stdlib.h"

void main()
{

    int a, b;
    float c;
    a = 1;
    b = 2;
    c = (float) a / b;

    printf("%f", c);

    system("PAUSE");
}
```

Correto!

Uso do #define

Aplicações:

- Constantes
- Macros

Uso do #define

Constantes

Exemplo:

```c
#include "stdio.h"
#include "stdlib.h"

#define PI 3.14159
#define DISCIPLINA "MTP"

void main ()
{
    printf("Disciplina: %s\n", DISCIPLINA);
    printf("O numero pi vale: %.5f\n",PI);
    system("PAUSE");
}
```

Macro

- ***Conceito:*** Na computação, MACRO é uma ***abstração*** que define como um padrão de ***entrada*** deve ser substituído por um padrão de ***saída***, de acordo com um conjunto de regras.

Uso do #define

Macros

Exemplo:

```c
#include "stdio.h"
#include "stdlib.h"

#define SOMA(a,b) ((a)+(b))

void main()
{
    float a, b, c;
    printf("Digite os valores de a e b: ");
    scanf("%f %f", &a, &b);
    c = SOMA(a, b);
    printf("O resultado e: %f\n", c);

    system("PAUSE");
}
```

Uso do #define e #undef

Sintaxe:

#define nome_da_macro sequencia_de_caracteres

#undef nome_da_macro

Uso do typedef

Exemplo 1:

```c
#include <stdio.h>
#include "stdlib.h"

#define MAX 50

void main()
{
    typedef struct
    {
        char nome[30];
        float salario;
        int tempodeServico;
    } Funcionario;

    Funcionario funcionarios[MAX];
}
```

Uso do typedef

Exemplo 2:

```c
#include <stdio.h>
#include "stdlib.h"

typedef int INTEIRO;
typedef float REAL;

void main()
{
   INTEIRO i;
   REAL a;

   i = 10;
   a = 2.34;

   printf("Os valores sao %d e %.2f\n", i, a);
   system("PAUSE");
}
```

Recursividade e Iteratividade

Exemplo : Cálculo do fatorial

Fatorial iterativo

Exemplo:

```c
#include "stdio.h"
#include "stdlib.h"

int fatorial (int n)
{
    int i;
    int res = 1;

    for (i = n; i>1; i--)
    {
        res = res * i;
    }

    return res;
}


void main()
{
    int num, resultado;
    printf("Digite o numero para calculo do fatorial: ");
    scanf("%d", &num);

    resultado = fatorial(num);

    printf("O valor do fatorial de %d é  %d.\n", num, resultado);
    system("PAUSE");
}
```

Fatorial recursivo

Exemplo:

```c
#include "stdio.h"
#include "stdlib.h"

int fatorial (int n)
{
    if (n==1)
    {
        return 1;
    }
    else
    {
        return ( n * fatorial(n-1) );
    }
}

void main()
{
    int num, resultado;
    printf("Digite o numero para calculo do fatorial: ");
    scanf("%d", &num);

    resultado = fatorial(num);

    printf("O valor do fatorial de %d é  %d.\n", num, resultado);
    system("PAUSE");
}
```

Exercícios

1) Dados dois vetores:

A (com 5 elementos) e **B** (com 8 elementos), faça um programa em linguagem C que imprima todos os elementos comuns aos dois vetores.

Exercícios

2) Considere um vetor de inteiros de 10 posições.

Faça um programa em linguagem C que leia do usuário os valores de cada uma das posições desse vetor e, em seguida, imprima:

a) A quantidade de números pares.
b) A quantidade de números múltiplos de 5.
c) O valor e posição do **maior** elemento.
d) O valor e posição do **menor** elemento.

Exercícios

3) Faça um programa em linguagem C para realizar um sorteio de números aleatórios dentro de uma faixa de valores.

Obs.: Utilize um mecanismo para inicializar o gerador de números pseudoaleatórios e compare os resultados.

Exercícios

4) Crie um programa para realizar o controle acadêmico de uma disciplina. O sistema deverá contemplar o armazenamento de informações como: nome do aluno, número de matrícula, nota da primeira prova (P1), nota da segunda prova (P2) e nota da terceira prova (P3).

Considere: **NotaFinal = P1 (30) + P2 (30) + P3 (40)**

a) Realize o cadastro de 10 alunos.
b) Imprima o nome dos alunos com maior nota em P1, P2 e P3.
c) Considerando-se o valor de no mínimo 60 pontos para aprovação, imprima a lista dos alunos aprovados na disciplina.