

PROGRAMAÇÃO DE COMPUTADORES V - TCC- 00.323

Modulo 7: Programação com funções predefinidas pela linguagem

Aura - Erick
aconci@ic.uff.br, erickr@id.uff.br

Roteiro

- ▶ Operadores de Atribuição
- ▶ Funções pré-definidas do C
 - ▶ Funções Matemáticas
- ▶ **Teorema geral da numeração posicional**
- ▶ Base octal e hexadecimal
- ▶ Funções pré-definidas do C
 - ▶ Funções para Tratamento de Strings
 - ▶ Outros formatos de impressão I/O
 - ▶ Como o computador representa “letras e outros símbolos”
- ▶ Exercícios
- ▶ Trabalho 4

Operadores de Atribuição

Operador	Exemplo	Descrição
=	$x = y$	A variável x recebe o valor da variável y
+=	$x += y$	Equivale a $x = x + y$
-=	$x -= y$	Equivale a $x = x - y$
*=	$x *= y$	Equivale a $x = x * y$
/=	$x /= y$	Equivale a $x = x / y$
%=	$x \% = y$	Equivale a $x = x \% y$



Funções Matemáticas - Usar #include <math.h>

Função	Exemplo	Comentário
ceil	ceil(x)	Arredonda o número real para cima. Exemplo ceil(3.2) é 4
cos	cos(x)	Cosseno de x (x em radianos)
exp	exp(x)	e elevado à potência x
fabs	fabs(x)	Valor absoluto de x
floor	floor(x)	Arredonda o número real para baixo. Exemplo floor(3.2) é 3
log	log(x)	Logaritmo natural de x
log10	log10(x)	Logaritmo decimal de x
pow	pow(x,y)	Calcula x elevado à potência y
sin	sin(x)	Seno de x
sqrt	sqrt(x)	Raiz quadrada de x
tan	tan(x)	Tangente de x



Exercício 1

Construa um programa em C que calcule a quantidade de latas de tinta necessárias e o custo para pintar um tanque cilíndrico de combustível, onde são fornecidos a altura e o raio do cilindro.

- ▶ a lata de tinta custa R\$20,00
- ▶ cada lata contém 5 litros
- ▶ cada litro de tinta pinta 3 metros quadrados.

$$\text{Área do cilindro} = 3.14 * \text{raio}^2 + 2 * 3.14 * \text{raio} * \text{altura}$$



Exercício 2

- ▶ Fazer um programa que pergunta as coordenadas de 2 pontos do plano e imprima **a distância Euclidiana se o usuário escolher E, ou distância a D entre eles se for essa a opção escolhida pelo usuário.**
- ▶ Como obter essas distâncias ao quadrado é comentado no módulo 3, logo para chegar a elas basta extrair a raiz quadrada , `sqrt(x)`, ou elevar esse valor a $\frac{1}{2}$: `pow(x,0.5)`



Exercício 3

- ▶ Calcule **N** pontos pertencentes a um círculo, de centro em **(x_0, y_0)** e raio **r** , sendo esses 4 parâmetros definido pelo usuário.
- ▶ Lembre-se que o círculo é o lugar geométrico dos pontos distantes **r** do centro, dados por **$ang \in [0, 2\pi)$**
- ▶ Ou **$x = x_0 + r \cos (2\pi \text{ ang } /N)$**
- ▶ **$y = y_0 + r \sin (2\pi \text{ ang } /N)$**

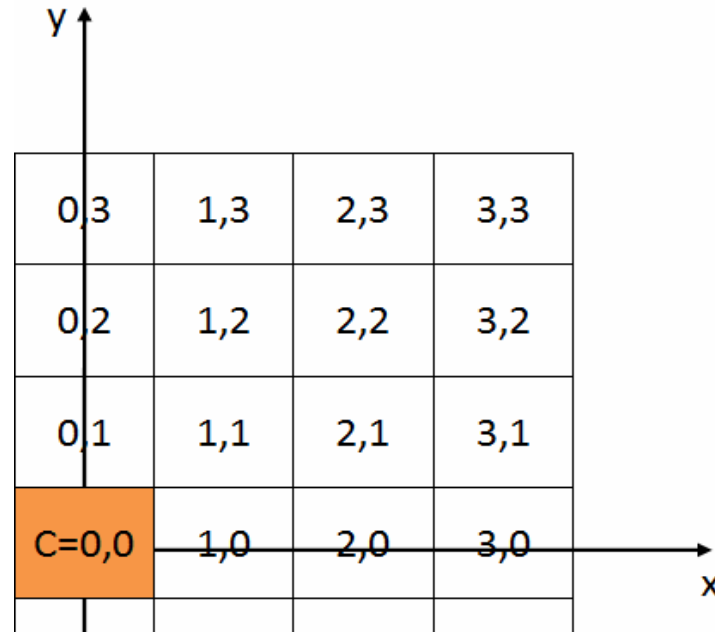


Trabalho 4:

- ▶ Faça o **fluxograma de** um programa que dependendo do usuário escolher **M** ou **S**, mostre na tela uma área com uma **Matriz** ou uma **Seqüência** de números.
- ▶ No caso do usuário escolher **M**, ou seja, no caso **da Matriz**, essa será calculada pelo preenchimento dos elementos de um array de 8 linhas e 8 colunas (8x8), com valores relacionados às posições (x , y) deste elemento dada pela expressão: $(x^r + y^r)^{1/r}$
- ▶ ***onde r é a ordem do seu nome na lista de chamada da turma.***



A figura acima mostra como seria uma parte destas posições (x , y).



Assim, por exemplo,

- ▶ se você for o **primeiro aluno**, $r=1$, alguns destes valores serão, os mostrados na figura

7							
6	7						
5	6	7					
4	5	6	7				
3	4	5	6	7			
2	3	4	5	6	7		
1	2	3	4	5	6	7	
0	1	2	3	4	5	6	7

Se você for o segundo aluno, $r=2$, alguns destes valores serão os mostrados na figura que segue:

3	$\sqrt{10}$	$\sqrt{13}$	$\sqrt{18}$
2	$\sqrt{5}$	$\sqrt{8}$	$\sqrt{13}$
1	$\sqrt{2}$	$\sqrt{5}$	$\sqrt{10}$
0	1	2	3

E assim por diante.*



No caso do usuário escolher **S**,

- ▶ ou seja, no caso da **Seqüência**, crie uma função no seu programa que imprime os elementos de ordem **$n=40$ a 50** de uma **seqüência de Lucas $U(n)$** : (como explicado no módulo 6), definida como:
- ▶ **$U(0) = 0$**
- ▶ **$U(1) = 1$**
- ▶ **$U(n+2) = P U(n+1) - Q U(n)$**
- ▶ onde **P** e **Q** são constantes definidas em função de **seu número na lista de chamada**, de forma que **P** é a soma dos dois dígitos que compõem seu número e **Q** é a diferença do primeiro dígito pelo segundo, desde que eles não sejam iguais. Caso sejam iguais então **$Q=1$** .



Por exemplo,

- ▶ se seu número for **01**, $P=1$ e $Q=0-1=-1$ de modo que $U(n+2) = U(n+1) + U(n)$ e a seqüência fica: **0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89.....** (que é a seqüência normal de *Fibonacci*, um dos casos particulares da de *Lucas*).
- ▶ **Mais exemplos:** Se seu número for **23**, $P=5$ e $Q=-1$, de modo que $U(n+2) = 5 U(n+1) + U(n)$.
- ▶ Se for **33**, $P=6$ e $Q=1$ de modo que $U(n+2) = 6 U(n+1) - U(n)$
- ▶ Se for **19**, $P=10$ e $Q=-8$ de modo que $U(n+2) = 10 U(n+1) + 8 U(n)$.



Depois de definir sua seqüência

- ▶ ***deve considerar como uma função poderia ser desenvolvida para fazer esse trabalho. E faça a impressão os valores de sequencia pedidos, isso é de U(40), U(41)U(50).***
- ▶ **Depois de feito o fluxograma e testado, (mostre como o testou), faça o código em C e rode o programa.**
- ▶ Entregue fluxograma, código e executável, até 22/01 (sexta mesmo!).
- ▶ Baseie-se no exemplo do calculo do Fatorial do modulo 6.
- ▶ Você pode encontrar pela internet também algumas soluções para o caso da seqüência de Fibonacci que pode seguir como dica, para desenvolver sua sequencia de Lucas.



teorema geral da numeração posicional

TGNP

- ▶ Define-se como sistema de numeração o ***conjunto de símbolos utilizados para a representação de quantidades e as regras que definem a forma de representação.***
 - ▶ Um sistema de numeração é determinado fundamentalmente pela **base**, que é o **número de símbolos utilizado**. A base é o coeficiente que determina qual o valor de cada símbolo de acordo com sua posição.
 - ▶ Os sistemas de numeração atuais são **sistemas posicionais**, em que o **valor relativo de cada símbolo** ou algarismo representa **depende do seu valor absoluto e da sua posição em relação a vírgula**. O valor relativo está diretamente ligado ao valor da base do sistema de numeração.
-



-
- A **base** do sistema decimal é o número **10**,
 - com a utilização dos seguintes símbolos:
 - **0 1 2 3 4 5 6 7 8 9.**



Exemplo qual o valor do 5?

- ▶ Base decimal:
 - ▶ 15
 - ▶ Valor absoluto = 5
 - ▶ Valor relativo = $5 \times 10^0 = 5$
- ▶ 51
 - ▶ Valor absoluto = 5
 - ▶ Valor relativo = $5 \times 10^1 = 50$



Outros sistemas

- ▶ ***O sistema octal***

- ▶ A **base** do sistema octal é o número **8**, com a utilização dos seguintes símbolos: **0 1 2 3 4 5 6 7**.

- ▶ ***O sistema hexadecimal***

- ▶ A **base** deste sistema é o número **16**, com a utilização dos seguintes símbolos: **0 1 2 3 4 5 6 7 8 9 A B C D E F**.
- ▶ Os valores absolutos de **A, B, C, D, E e F** são, respectivamente, **10, 11, 12, 13, 14 e 15**.
- ▶ Podem ser também **a ou A, b ou B, c ou C, d ou D, e ou E e f ou F**



Exemplo qual o valor do 5?

- ▶ Base octal:
 - ▶ 15
 - ▶ Valor absoluto = 5
 - ▶ Valor relativo = $5 \times 8^0 = 5$
 - ▶ 51
 - ▶ Valor absoluto = 5
 - ▶ Valor relativo = $5 \times 8^1 = 40$



Exemplo qual o valor do 5?

- ▶ Base hexadecimal:
- ▶ 15
- ▶ Valor absoluto = 5
- ▶ Valor relativo = $5 \times 16^0 = 5$
- ▶ 51
- ▶ Valor absoluto = 5
- ▶ Valor relativo = $5 \times 16^1 = 80$



Base Oito

- ▶ **Sistema Octal** tem base é 8, ou seja, utiliza 8 símbolos para a representação de quantidade.
- ▶ Estes símbolos são : 0 1 2 3 4 5 6 7
- ▶ O octal foi muito utilizado em informática como uma **alternativa mais compacta que o binário**.
- ▶ O hexadecimal, hoje, é mais utilizado como alternativa ao binário.
- ▶ Este sistema também é um posicional: a posição de seus determina o valor.
- ▶ Cada deslocamento para a esquerda, multiplica o valor por 8, e o divide por 8 se à direita (mesmo se depois da virgula, ou ponto, do número).
- ▶ A aritmética é semelhante a dos sistemas decimal e binário, motivo pelo qual não será apresentada, **só cobrada**



Exemplo:

- ▶ Qual o número decimal representado pelo número octal 4701?
- ▶ Utilizar o TGNP :
 - $4701 = 4 \times 8^3 + 7 \times 8^2 + 0 \times 8^1 + 1 \times 8^0 =$
 - $4701 = 4 \times 512 + 7 \times 64 + 0 + 1 =$
 - $4701 = 2048 + 448 + 0 + 1 = 2497$



Outro Exemplo:

- ▶ Converter o número octal 764 para o sistema decimal
- ▶ $764_{(8)} = 7 \times 8^2 + 6 \times 8^1 + 4 \times 8^0 =$
- ▶ $764_{(8)} = 7 \times 64 + 6 \times 8 + 4 \times 1 =$
- ▶ $764_{(8)} = 448 + 48 + 4 =$
- ▶ $764_{(8)} = 500_{(10)}$



O sistema binário

- ▶ A **base** do sistema binário é o número **2**,
- ▶ com a utilização dos seguintes símbolos:
- ▶ **0 e 1 (Binary digit)**.

- ▶ É o sistema de numeração utilizado pelos computadores devido sua fácil representação os circuitos eletrônicos, memórias, etc..



Tabela de conversão de bases

N.º Decimal 10	N.º Binário 2	N.º Hexadecimal 16	N.º Octal 8
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	8	10
9	1001	9	11
10	1010	A	12
11	1011	B	13
12	1100	C	14
13	1101	D	15
14	1110	E	16
15	1111	F	17



Conversão Octal – Binário

- ▶ Quando existir necessidade de converter números octal em binários, deve-se separar cada dígito do número octal e substituí-lo pelo seu valor correspondente de binário.
- ▶ Exemplo:
 - Converter o número octal 1572 em binário.
 - $1\ 5\ 7\ 2 = 001\ 101\ 111\ 010$



-
- ▶ Determinados conjuntos de dígitos binários (bits) possuem um nome especial:
 - ▶ **Quarteto** = 4 bits
 - ▶ **Octeto** ou **Byte** = 8bits
 - ▶ **Kilobyte** = 1024 bytes
 - ▶ **Megabyte** = 1024 Kbytes
 - ▶ **Gigabyte** = 1024 Mbytes
 - ▶ **Terabyte** = 1024 Gbytes



Conversão Binário – Octal

- ▶ Para converter um número binário em octal, executa-se o processo inverso ao anterior:
- ▶ Agrupam-se **os dígitos binários de 3 em 3**, substituindo-se cada trio de dígitos binários pelo equivalente dígito octal.
- ▶ Por exemplo, a conversão do número binário 1010111100 em octal:

001	010	111	100
1	2	7	4

Assim, tem-se $1010111100_{\text{bin}} = 1274_{\text{oct}}$

Conversão de Binário para Hexadecimal

- ▶ Um dígito em hexadecimal pode representar um número binário de 4 dígitos, dessa forma, para transformar um binário em hexadecimal, separamos **o binário em grupos de 4 bits**, começando pela direita.

Exemplo:

Binário: 1101000101100011.

1º - separar em grupos de quatro bits:

```
1101 0001 0110 0011
```

2º - identificar os números hexadecimais correspondentes:

```
1101 = D  
0001 = 1  
0110 = 6  
0011 = 3
```

Hexadecimal: D163.



Conversão de Hexadecimal para Binário

- ▶ É o inverso do processo anterior.
- ▶ Cada dígito será transformado em um número binário de 4 bits.

Exemplo:

Binário: 1101000101100011.

1º - separar em grupos de quatro bits:

```
1101 0001 0110 0011
```

2º - identificar os números hexadecimais correspondentes:

```
1101 = D  
0001 = 1  
0110 = 6  
0011 = 3
```

Hexadecimal: D163.

entrada e saída (*input e output*).

- ▶ um computador é praticamente inútil se não tiver nenhuma maneira de interagir com o usuário.
- ▶ As trocas de informação entre o computador e o usuário são chamadas **entrada e saída**
- ▶ **Entrada** é a informação fornecida a um programa;
- ▶ **Saída** é a informação fornecida pelo programa.
- ▶ refere-se aos dois termos simultaneamente como : *entrada/saída* ou **E/S (I/O)**.
- ▶ "saída padrão" (standard output, stdout) e "entrada padrão" (standard input, stdin), se referem, na maioria das vezes, **ao monitor e ao teclado**, que são os meios básicos de interação com o usuário.



Funções para tratamento de Strings

- ▶ Não existe um tipo String em C
- ▶ Strings em C são um vetor do tipo char que termina com '\0'
- ▶ Para literais String, o próprio compilador coloca '\0'

```
#include <stdio.h>
#include <stdlib.h>
main() {
    char ch[8] = "lagarto";
    printf ("%s\n", ch);
    system("pause");
}
```



Ler uma String: **gets**

▶ Comando **gets**

```
#include <stdio.h>
#include <stdlib.h>
main() {
    char ch[80];
    printf ("Digite o seu nome: ");
    gets(ch);
    printf ("Oi %s\n", ch);
    system("pause");
}
```



gets × scanf

- ▶ Pode-se utilizar scanf no lugar do gets, mas...
 - ▶ scanf não lê espaços em branco.
- ▶ Se o usuário digitar: “Alexandre Costa e Silva”
 - ▶ o scanf vai ler apenas “Alexandre”
 - ▶ o gets vai ler tudo



Comparar duas strings: **strcmp**

- ▶ `strcmp (s1, s2);`
- ▶ `strcmp` retorna 0 se as duas strings são iguais.
- ▶ Precisa do `#include <string.h>`



Exemplo

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
main( ) {
    char ch[80];
    printf ("Digite a senha: ");
    scanf ("%s", &ch);
    if (strcmp(ch,"laranja") == 0)
        printf ("Senha correta\n");
    else printf ("Senha invalida\n");
    system("pause");
}
```



Tamanho de uma string: **strlen**

- ▶ `int size = strlen(str);`
 - ▶ Retorna um valor inteiro com o número de caracteres da String.



Exemplo

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
main(){
    char ch[80];
    printf ("Digite a palavra: ");
    scanf ("%s", &ch);
    int size = strlen(ch);
    printf ("Esta palavra tem %d caracteres.\n", size);
    system("pause");
}
```



Copiar conteúdo de uma string: **strcpy**

▶ strcpy(para, de)

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
main()
{
    char str[80];
    strcpy (str, "Alo");
    printf ("%s \n", str);
    system("pause");
}
```



Concatenar duas strings: strcat

- ▶ Concatenação = junção de duas strings
- ▶ `strcat(str1, str2)`
 - ▶ Concatena a `str1` com a `str2`, colocando o resultado em `str1`

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
main() {
    char palavra1[30], palavra2[30];
    printf ("\n Informe uma string: ");
    gets (palavra1);
    printf ("\n Informe uma segunda string: ");
    gets (palavra2);
    strcat (palavra1,palavra2);
    printf ("A palavra concatenada eh %s \n", palavra1);
    system("pause");
}
```



Exemplo com várias funções

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
main() {
    char palavra[30], palavra2[30], teste[30];
    int comp;
    printf ("\n Informe uma string: ");
    gets (palavra);
    printf ("A palavra eh %s\n", palavra);
    printf ("\n Informe uma segunda string: ");
    gets (palavra2);
    printf ("A palavra eh %s\n", palavra2);
    strcpy (teste, "aula teste");
    printf ("A palavra copiada eh %s\n", teste);
    strcat (teste, " 1");
    printf ("A palavra concatenada eh %s\n", teste);
    printf ("\nO tamanho da primeira string eh %d",
strlen(palavra));
    comp = strcmp (palavra, palavra2);
    if (comp==0) printf ("\nSao iguais: %d", comp);
    else printf ("\nSao diferentes: %d", comp);
    printf ("\nMaiusculo: %s", strupr(palavra));
    printf ("\nMinusculo: %s\n", strlwr(palavra));
    system("pause");
}
```


Exercício 1

Escreva um programa que leia duas strings e as imprima na tela. Imprima também a segunda letra de cada string.



Exercício 2

Escreva um programa que leia uma string, conte quantos caracteres desta string são iguais a 'a' e substitua os que forem iguais a 'a' por 'b'. O programa deve imprimir o número de caracteres modificados e a string modificada.



Exercício 3

Faca um programa que leia o nome de 5 pessoas e mostre os nomes armazenados sempre em Maiúsculas. Utilize vetores.



Exercício 4

Uma empresa concederá um aumento de salário aos seus funcionários, variável de acordo com o cargo, conforme a tabela abaixo. Faça um programa que:

- ▶ Leia o salário de um funcionário
- ▶ Leia o cargo de um funcionário
- ▶ Calcule o novo salário

Se o cargo do funcionário não estiver na tabela, ele deverá então receber 40% de aumento. Mostre o salário antigo, o novo salário e a diferença.

Cargo	Percentual
Gerente	10%
Engenheiro	20%
Técnico	30%



Exercício - Resposta

Escreva um programa que leia duas strings e as imprima na tela. Imprima também a segunda letra de cada string.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
main(){
    char palavra[30];
    printf ("\n Informe uma string: ");
    gets (palavra);
    printf ("A palavra digitada eh %s\n", palavra);
    printf ("\nA segunda letra da string digitada eh %c", palavra[1]);
    printf ("\n Informe uma segunda string: ");
    gets (palavra);
    printf ("A palavra digitada eh %s\n", palavra);
    printf ("\nA segunda letra da string digitada eh %c\n", palavra[1]);
    system("pause");
}
```



Exercício

Escreva um programa que leia uma string, conte quantos caracteres desta string são iguais a 'a' e substitua os que forem iguais a 'a' por 'b'. O programa deve imprimir o número de caracteres modificados e a string modificada.



```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
main(){
    char string[30];
    int tamanho, i, contaA=0;
    printf ("\nInforme uma string: ");
    gets (string);
    tamanho=strlen(string);
    for (i=0;i<tamanho;i++)    {
        if (string[i]=='a')    {
            contaA++;
            string[i]='b';
        }
    }
    printf ("\nO numero de caracteres modificados eh %d", contaA);
    printf ("\nA string modificada eh %s\n", string);
    system("pause");
}
```

Exercício - Resposta

Exercício - Resposta

Faca um programa que leia o nome de 5 pessoas e mostre os nomes armazenados. Utilize vetores.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
main(){
    char nomes[5][10];
    int cont;
    for (cont=0;cont<5;cont++)
    {
        printf ("\nInforme o nome %d: ", cont);
        gets(nomes[cont]);
    }
    for (cont=0;cont<5;cont++)
        printf ("\nO nome armazenado na posicao %d eh %s", cont, nomes[cont]);
    system("pause");
}
```



Exercício – Resposta

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
main(){
    char cargo[20];
    float salario, novoSalario;
    printf ("\nDigite o cargo: ");
    scanf ("%s", cargo);
    printf ("\nDigite o salario: ");
    scanf ("%f", &salario);
    if (strcmp(cargo, "gerente")==0)
        novoSalario=salario*1.1;
    else if (strcmp(cargo, "engenheiro")==0)
        novoSalario=salario*1.2;
    else if (strcmp(cargo, "tecnico")==0)
        novoSalario=salario*1.3;
    else novoSalario=salario*1.4;
    printf ("\nO salario antigo eh %.2f, o salario novo eh %.2f e a diferenca eh
de R$ %.2f\n", salario, novoSalario, novoSalario-salario);
    system("pause");
}
```

Como multiplicar seu aprendizado nos Exercícios

- ▶ Fazer rodar cada um dos exercícios na sua máquina.
- ▶ Sugerir baseado neles alterações dos exercícios, e as enviar a um colega.
- ▶ (por exemplo novo 3: Faça um programa que leia o nome de 5 pessoas e mostre o último sobrenome sempre em Maiúsculas).
- ▶ Verifique se as respostas do seu colega estão corretas, executando-as em sua máquina.



Entrada e saída em C:

- ▶ as funções da biblioteca padrão para entrada e saída estão declaradas no cabeçalho **stdio.h**
- ▶ A mais usada já foi introduzida em seções anteriores: **printf()**
- ▶ **printf** vem de "print formatted" (imprimir formatado).
- ▶ Ela permite facilmente imprimir valores que não são sequências de caracteres, além de poder formatar os dados e juntar várias sequências de caracteres. Por isso, a função `printf()` é a mais usada !!



Exemplos:

- ▶ `printf("Este é um programa em C");`
- ▶ `printf("Ola" " mundo" "!!!");`
- ▶ `int teste = 42;`
- ▶ `printf ("A variável 'teste' é o número %d. A variável`

□ `'teste' contém o número 42. ", teste);`

```
float pi = 3.1415; printf ("O valor de pi é %f.", pi);
```

O valor de pi é 3.1415.



Especificações de formato

- %d Número decimal inteiro (int).
 - %u Número decimal natural (unsigned int), ou seja, sem sinal.
 - %o Número inteiro representado na base octal.
 - Exemplo: 41367 (corresponde ao decimal 17143).
 - %x Número inteiro representado na base hexadecimal.
 - Exemplo: 42f7 (corresponde ao decimal 17143).
 - %X idem a %x mas com as letras serão maiúsculas:
 - Exemplo: 42F7.
 - %f Número decimal de ponto flutuante.
 - %lf **double**.
 - %e Número em notação científica, por exemplo 5.97e-12.
 - %E Número em notação científica em maiúsculo (5.97E-12).
 - %g Escolhe automaticamente o mais apropriado entre %f e %e.
 - %G escolhe 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 ao valor dado.
 - %s Sequência de caracteres (string, em inglês).
 - %% Imprime um %
-

