

Entrada formatada de dados: função scanf

Sintaxe:

```
scanf ("formatos", &var1, var2,...);
```

Exemplos:

```
int i, j;
float x;
char c;
scanf("%d", &i);
scanf("%d%f", &j, &x);
scanf("%c", &c);
```

```
%d 1 número inteiro
%i mesmo que %d
%f 1 número float
%lf 1 número double
%c 1 caractere char
```

Slide 1

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Saída formatada de dados: função printf

Sintaxe:

```
printf ("formatos", var1, var2,...);
```

Exemplos:

```
int i, j;
float x;
char c;
printf("Resultados: i=%d \n
j=%d \n", i, j);
printf("%d", i);
printf("%d, %f", j, x);
printf("%c", c);
```

```
%c char
%d inteiro
%i inteiro
%f float ou double, notação
comum - 6 casas decimais
%e float ou double com
notação científica
dependendo do
conteúdo, usa %f - 3 casas
decimais - ou %e
```

	Função
\b	BackSpace
\n	New Line (mudança de Linha)
\t	Tabulação Horizontal
\v	Tabulação Vertical
\	imprime o próprio caractere \
'\'	imprime o caractere '
"\"	imprime o caractere "
\?	imprime o caractere ?
%%	imprime caractere %

Slide 2

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Comando (operador) de atribuição

Sintaxe:

```
identificador = expressão;
```

- Atribui o valor da direita à variável da esquerda
- O valor pode ser uma constante, uma variável ou uma expressão

Exemplos

```
x = 4; // significa que a variável X recebe conteúdo 4
y = x + 2;
y = y + 4;
valor = 2.5; //casa decimal indicada pelo ponto
sexo = 'F'; /*atribuição de 1 único caractere para uma
variável char é entre apóstrofes*/
```

Slide 3

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Operadores Aritméticos

Operador	Exemplo	Comentário
+	x + y	Soma x e y
-	x - y	Subtrai y de x
*	x * y	Multiplica x e y
/	x / y	Divide x por y
%	x % y	Resto da divisão de x por y
++	x++	Incrementa em 1 o valor de x
--	x--	Decrementa em 1 o valor de x

Slide 4

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Operadores de Atribuição (atalhos)

Operador	Exemplo	Comentário
=	x = y	Atribui o valor de y a x
+=	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

Slide 5

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

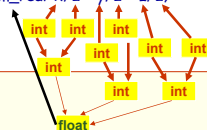
Modeladores - casts

Sintaxe:

```
(tipo) expressão
```

- Altera o tipo retornado por operações parciais de uma expressão.
- Funcionamento:

```
float num_real; // variável real
int x=1,y=2,z=1; // valores inteiros - inicializados na declaração
//expressão com operadores inteiros,todos resultados parciais em
inteiros:
num_real=x/2 + y/2 + z/2;
```



0 + 1 + 0 ↔ 1

Slide 6

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Modeladores em expressões aritméticas

```

/*testa cast (modeladores) nos operandos*/
#include <stdio.h>
#include <stdlib.h>
int main()
{
    float num_real; // variável real
    int x=1,y=2,z=1; // valores inteiros - inicializ
    system("color f3"); // modifiquei cores do
    //expressão com operadores inteiros,todos resultados parciais em inteiro:
    num_real=x/2 + y/2 + z/2;
    printf("Caso 1: %f\n",num_real);
    //operadores inteiros, 1o resultado parcial em real
    num_real=(float)x/2 + y/2 + z/2; //operadores inteiros, 1 resultado parcial em real
    printf("Caso 2: %f\n",num_real);
    //operadores inteiros,todos resultados parciais em real
    num_real=(float)x/2 + (float)y/2 + (float)z/2;
    printf("Caso 3: %f\n",num_real);
    //expressão com operadores inteiros e reais, todos resultados parciais em real;
    num_real = x / 2.0 + y / 2.0 + z / 2.0; // inserção de espaços : + legibilidade
    printf("Caso 4: %f\n",num_real);
    system("pause");
    return 0;
}

```

Slide 7

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Funções Matemáticas (#include <math.h>)

Função	Exemplo	Comentário
ceil	ceil(x)	Arredonda o número real para cima: 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: 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

Slide 8

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Operadores lógicos ou booleanos

Operador	Tipo	Significado
&&	binário	"e"
	binário	"ou"
!	unário	negação

&&	V	F
V	V	F
F	F	F

	V	F
V	V	V
F	V	F

!	V	F
F	V	F

Em C, o **valor** das expressões lógicas são:

- 0, se falso;
- 1 ou outro valor que **não zero**, se verdadeiro.

Slide 9

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Operadores Relacionais

Operador	Exemplo	Comentário
==	x == y	O conteúdo de x é igual ao de y
!=	x != y	O conteúdo de x é diferente do de y
<=	x <= y	O conteúdo de x é menor ou igual ao de y
>=	x >= y	O conteúdo de x é maior ou igual ao de y
<	x < y	O conteúdo de x é menor que o de y
>	x > y	O conteúdo de x é maior que o de y

Em C, o **valor** das expressões relacionais são:

- 0, se falso;
- 1 ou outro valor que **não zero**, se verdadeiro.

Slide 10

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Operadores Relacionais

A é maior que 10? (a > 10)

X é negativo? (x < 0)

y é diferente de 10? (y != 10)

nota é igual a 5? (nota == 5)

Slide 11

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

A expressão lógica em C

Sintaxe:

```

<operando> <operador booleano> <operando>
ou
<operando> <operador relacional> <operando>

```

- Retorna um valor lógico:
 - falso: zero;
 - verdadeiro: 1 (ou qualquer valor diferente de zero).
- Operandos podem ser expressões aritméticas ou lógicas, incluindo operadores aritméticos, relacionais e lógicos.
- Exemplos:


```

(a > b) || (c < d)
valor > 10
!(a+c<b)
(a == b)
(a==b) && (a==c)
(a!=b)
!((a==b) || (a+c)) // se resultado da soma != 0, considera V!

```

Slide 12

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Expressões lógicas ou relacionais

a) código entre 1 e 100 (valores limites considerados):

(código > 0 && código < 101)

(0 < código < 101)

C:\Cora\Disciplinas\INF01202 CVP

```
codigo = -1
<codigo > 0 && codigo < 101> false
<0 < codigo < 101> verdadeiro!

codigo = 5
<codigo > 0 && codigo < 101> verdadeiro
<0 < codigo < 101> verdadeiro!

codigo = 101
<codigo > 0 && codigo < 101> false
<0 < codigo < 101> verdadeiro!

Pressione qualquer tecla para continuar. . .
```

Slide 13

Informática

Precedência de operadores em C

Na expressão: (x != 10 || y > 1 && y < 10)

1. Pela precedência dos operadores, primeiro serão resolvidos ">" e "<";
y > 1 e y < 10
2. O próximo operador a ser considerado, pela precedência será "!="
x != 10
3. Restam os operadores "&&" e "||", como "&&" é o de maior prioridade, será o próximo:
y < 1 && y < 10
4. Finalmente o "||" será resolvido.

A versão a seguir com parênteses mostra a ordem de execução da expressão:
((x != 10) || ((y > 1) && (y < 10)))

RECOMENDAÇÃO: usar parênteses não só para alterar a precedência de operadores, como também para tornar mais claras as expressões.

Slide 14

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Expressões lógicas ou relacionais

b) contador maior ou igual a maximo ou valor igual a procurado:

(contador >= maximo || valor == procurado)

c) valor igual a 1 ou 2 ou entre 5 e 7 (inclusive):

(valor == 1 || valor == 2 || (valor >= 5 && valor <= 7))

Slide 15

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Comandos de seleção condicional

→ Condicional Simples

```
if (condição)
    comando;
```

• Seleção Dupla

```
if (condição)
    comando;
else comando;
```

• Seleção múltipla

```
switch (variável)
{
    expressão;
}
```

Slide 16

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

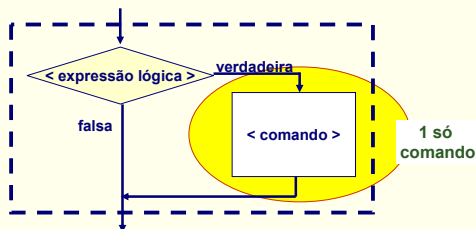
Comando condicional simples

C

```
if ( expressão lógica )
    comando ;
```

Linguagem algorítmica:

```
se < expressão lógica >
    < comando >
```



Slide 17

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Comando: simples / composto

Comando simples

```
comando;
```

1 só comando

Comando composto

```
{
    < comandos, separados por ponto e vírgula >
}
```

1 comando (bloco)

Slide 18

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

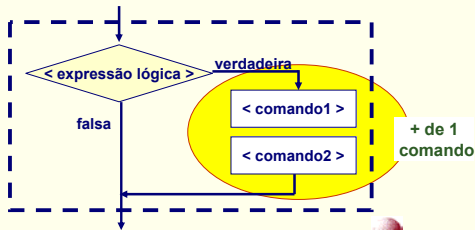
Comando condicional simples

C

```
if ( expressão lógica )
{
    comando1;
    comando2;
}
```

Linguagem algorítmica:

```
se < expressão lógica >
    < comandoS >
```



Slide 19

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Comando condicional simples

Ex1: Ler um valor e, no caso de ser menor do que 10, emitir uma mensagem.

Algoritmo:

1. início.
2. ler valor.
3. se valor < 10
- 3.1 escrever mensagem
4. fim.

```
/* Lê valor e informa se maior que 10 */
#include <stdio.h>
#include <stdlib.h>

int main() // programa principal
{
    int val1; // declaração da variável informada
    printf("Valor 1:\n"); // mensagem impressa na tela
    scanf("%d", &val1); // leitura da informação digitada
    if (val1 < 10)
        printf("%d é menor que 10\n", val1); // mensagem
    system("pause");
    return 0;
}
```

Slide 20

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Comando condicional simples

Ex2: Processar uma venda de livros em uma livraria. Obter código do tipo de livro vendido (A, B, C) e número de unidades, calcular e informar valor a pagar.

Preços: Tipo A: R\$ 10,00
Tipo B: R\$ 20,00
Tipo C: R\$ 30,00

Caso tenham sido vendidas mais de 10 livros, emitir uma mensagem.

Algoritmo UmaVenda

{processa uma venda e avisa caso tenham sido vendidas mais de 10 unidades}
entradas: código {do tipo do livro}
nro_livros {vendidos}
saídas: apagar
mensagem {caso tenham sido vendidas mais de 10 unidades}

1. início
2. ler código, nro_livros
3. calcular preço a pagar
4. informar apagar
5. se nro_livros > 10
- 5.1 informar "mais de 10 livros vendidos"
6. fim

```
se código = 'A'
    apagar ← nro_livros * 10;
se código = 'B'
    apagar ← nro_livros * 20;
se código = 'C'
    apagar ← nro_livros * 30;
```

Slide 21

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Ex2: Processar uma venda de livros em uma livraria. Obter código do tipo de livro vendido (A, B, C) e número de unidades. Preços: Tipo A: R\$ 10,00
Tipo B: R\$ 20,00
Tipo C: R\$ 30,00
Calcular e informar preço a pagar. Caso tenham sido vendidas mais de 10 livros, emitir uma mensagem.

Algoritmo UmaVenda

{processa uma venda e avisa caso tenham sido vendidas mais de 10 unidades}
entradas: código {do tipo do livro}
nro_livros {vendidos}
saídas: apagar
mensagem {caso tenham sido vendidas mais de 10 unidades}

1. início
2. ler código, nro_livros
3. se código = 'A'
- 3.1 apagar ← nro_livros * 10;
4. se código = 'B'
- 4.1 apagar ← nro_livros * 20;
5. se código = 'C'
- 5.1 apagar ← nro_livros * 30;
6. informar apagar
7. se nro_livros > 10
- 7.1 informar "mais de 10 livros vendidos"
8. fim

Slide 22

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

```
/* processa uma venda e avisa quando mais de
10 unidades foram vendidas */
#include <stdio.h>
#include <stdlib.h>

int main()
{
    char código;
    int nro_livros;
    float apagar;
    printf("digite o código do produto e número de unidades:\n");
    scanf("%c %d", &código, &nro_livros); //sem espaços entre formatação!
    if (código == 'a' || código == 'A')
        apagar = nro_livros * 10;
    if (código == 'b' || código == 'B')
        apagar = nro_livros * 20;
    if (código == 'c' || código == 'C')
        apagar = nro_livros * 30;
    printf("o valor a pagar é R$ %.2f\n", apagar); //valor real: formatar!!
    if (nro_livros > 10)
        printf("Foram vendidas mais de 10 unidades do tipo %c", código);
    system("pause");
    return (0);
}
```

Slide 23

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Fazer agora: localização de um ponto no plano

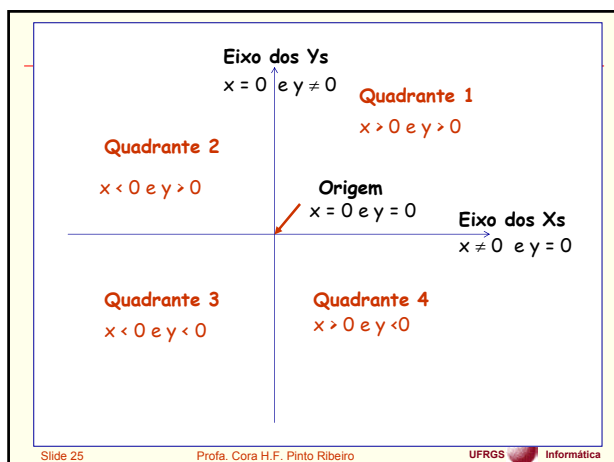
Dados um par de valores x e y, que representam as coordenadas de um ponto no plano, determinar a localização do ponto: se em um quadrante, em um dos eixos ou na origem.

- **Entradas:** coordenadas x e y de um ponto.
- **Processamento:** testes, verificando onde o par de valores se enquadra.
- **Saída:** mensagem adequada.

Slide 24

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática



Slide 25

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Passos do Algoritmo

- leitura dos valores de x e y .
- determinação, pela avaliação de condições, de onde o ponto se encontra: se em um quadrante, em um eixo ou na origem.
- escrita da mensagem, onde é indicada a localização do ponto - apenas 1 mensagem por par de valores.

Slide 26

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Escrita do algoritmo em pseudolinguagem

Algoritmo LocalizaQuadrante
 Entradas: x , y (coordenadas)
 Saída: mensagem (localização)

1. início
2. ler x , y
3. se $x = 0$ e $y = 0$
 escrever "ponto na origem"
4. se $x > 0$ e $y > 0$
 escrever "quadrante 1"
5. etc...
- (...)
9. se $x \neq 0$ e $y = 0$
 escrever "eixo dos x "
10. fim

Slide 27

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Em C:

```
/* Obtem coordenadas e informa localização */
#include <stdio.h>
int main ( )
{
    float x, y; // pontos são valores reais
    system ("color f9"); // fundo branco, letra azul
    printf("In Coordenadas: x = ");
    scanf("%f", &x);
    printf(" e y = ");
    scanf("%f", &y);
    if ( x == 0 && y == 0 )
        printf("In Ponto na origem");
    if ( x > 0 && y > 0 )
        printf("In Ponto no quadrante 1");
    if ( x < 0 && y > 0 )
        printf("In Ponto no quadrante 2");
    if ( x < 0 && y < 0 )
        printf("In Ponto no quadrante 3");
    if ( x > 0 && y < 0 )
        printf("In Ponto no quadrante 4");
    if ( x == 0 && y != 0 )
        printf("In Ponto no eixo dos y");
    if ( x != 0 && y == 0 )
        printf("In Ponto no eixo dos x");
    printf("\n");
    system("pause");
    return 0;
}
```

Slide 28

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática