

Funções de Entrada e Saída Formatada

Constantes de barra invertida

Constante	Significado
<code>\n</code>	new line
<code>\"</code>	aspas
<code>\'</code>	apóstrofo
<code>\0</code>	- nulo (0 decimal)
<code>\\</code>	barra invertida
<code>\t</code>	tabulação horizontal (tab)
<code>\a</code>	alerta (beep)
<code>\b</code>	retorno do cursor

Funções de Entrada e Saída Formatada

➔ scanf

➔ leitura de dados;

➔ formato geral:

scanf (string_de_controle, lista_de_argumentos);

➔ *string_de_controle* → descrição de todas as variáveis que serão lidas, com informações de seus tipos e da ordem em que serão lidas.

➔ *lista_de_argumentos* → lista com os identificadores das variáveis que serão lidas;

Funções de Entrada e Saída Formatada

➔ scanf (continuação)

- ➔ colocar antes de cada identificador da *lista_de_argumentos* o caracter '&'

Exemplo: char ch;
 scanf ("%c", &ch);

Funções de Entrada e Saída Formatada

➔ scanf (continuação)

Tabela de códigos de formato (%)

Código	Formato
%c	Um caracter (char)
%d	Um número inteiro decimal (int)
%f	Ponto flutuante decimal
%s	String

Funções de Entrada e Saída Formatada

Exercício:

Construa um programa em C que receba através da entrada padrão um valor inteiro pertencente ao intervalo $[0,255]$ e a posição do bit que se deseja saber o valor, o programa deve escrever na saída padrão o valor o bit solicitado.

Funções de Entrada e Saída Formatada

```
#include <stdio.h>
main ()
{
    unsigned char desloca, valor_byte, aux=1;
    printf("\nDigite um numero pertencente ao intervalo
    [0,255]: ");
    scanf("%d", &valor_byte);
    printf("\nDigite o bit a testar(0 a 7): ");
    scanf("%d", &desloca);
    aux = aux << desloca;
    valor_byte = valor_byte & aux;
    valor_byte = valor_byte >> desloca;
    printf("\nO valor do bit eh %d",valor_byte);
}
```

Estruturas de Controle de Fluxo

1. Instrução condicional

```
if (<condição>
    <instrução1>
```

Estruturas de Controle de Fluxo

1. Instrução condicional (continuação)

```
if (<condição>
    <instrução1>
else
    <instrução2>
```


Estruturas de Controle de Fluxo

```
if (<condição>
{
    <instrução1>
    .
    .
    <instruçãon>
}
else
{
    <instrução1>
    .
    .
    <instruçãom>
}
```

Estruturas de Controle de Fluxo

Exemplo:

```
#include <stdio.h>
main()
{
    float n1, n2, n3, n4, media;
    printf ("Entre com a primeira nota do aluno: ");
    scanf ("%f",&n1);
    printf ("\nEntre com a segunda nota do aluno: ");
    scanf ("%f",&n2);
    printf ("\nEntre com a terceira nota do aluno: ");
    scanf ("%f",&n3);
    printf ("\nEntre com a quarta nota do aluno: ");
    scanf ("%f",&n4);
    media=(n1+n2+n3+n4)/4;
    if (media>=7)
        printf ("\nAluno aprovado!");
    else
        printf ("\nAluno reprovado!");
```

112 }

Estruturas de Controle de Fluxo

1. Instrução condicional (continuação)

Exercício 1:

Construa um programa que calcule o salário líquido de um funcionário. O programa recebe através do teclado o salário bruto de um funcionário, caso este seja inferior a R\$ 300,00 é descontado 5% em impostos, se o salário estiver entre R\$ 300,00 e 1.200,00 é descontado 10% em impostos, se esse for superior a R\$ 1.200,00 é descontado 15% em impostos. O programa retornará no monitor o salário líquido do funcionário.

Estruturas de Controle de Fluxo

1. Instrução condicional (continuação)

Exercício 2:

Construa um programa em C que receba através da entrada padrão um valor inteiro pertencente ao intervalo $[0,255]$ e a posição do bit que se deseja verificar se está ativo, o programa deve indicar na saída padrão o resultado desta verificação.

Estruturas de Controle de Fluxo

2. Operador condicional

<expressão_lógica>?<valor1>:<valor2>

ou

<expressão_lógica>?<expressão1>:<expressão2>

Estruturas de Controle de Fluxo

2. Operador condicional (continuação)

Exemplo: O programa abaixo recebe dois inteiros através da entrada padrão e retorna na tela o inteiro com maior valor dentre os dois fornecidos.

```
#include <stdio.h>  
main()  
{  
    int a,b;  
    printf ("Entre com o valor de a: ");  
    scanf ("%d",&a);  
    printf ("\nEntre com o valor de b: ");  
    scanf ("%d",&b);  
    printf ("\nO maior valor e: %d",a>b?a:b);  
}
```

Estruturas de Controle de Fluxo

3. Laços de repetição

```
while (<condição>)
```

```
{
```

```
    <instrução1>
```

```
    .
```

```
    .
```

```
    .
```

```
    <instruçãon>
```

```
}
```

Estruturas de Controle de Fluxo

3. Laços de repetição (continuação)

Exemplo:

Dada uma reta $ax+by+c=0$ e cinco pontos, faça um programa para calcular, para cada ponto, o seguinte: se o ponto estiver no primeiro quadrante calcule e informe a distância do ponto a reta caso contrário escreva uma mensagem informando que o ponto não pertence ao primeiro quadrante.

Estruturas de Controle de Fluxo

```
#include <stdio.h>
#include <math.h>
main()
{
    float a,b,c,x,y;
    int contador=1;
    printf ("Equação da reta: ax+by+c=0\n");
    printf ("Coeficiente a da reta = ");
    scanf ("%f",&a);
    printf ("\nCoeficiente b da reta = ");
    scanf ("%f",&b);
    printf ("\nCoeficiente c da reta = ");
    scanf ("%f",&c);
```

```

while (contador<=5)
{
    printf ("\nCoordenada x do ponto = ");
    scanf ("%f",&x);
    printf ("\nCoordenada y do ponto = ");
    scanf ("%f",&y);
    if (x>=0.0 && y>=0)
        printf ("\nA distancia do ponto a reta eh: %f",
            fabs(a*x+b*y+c)/sqrt(pow(a,2)+pow(b,2)));
    else
        printf ("\nO ponto nao esta no primeiro
quadrante!");
    contador++;
}
}

```

Estruturas de Controle de Fluxo

3. Laços de repetição (continuação)

```
do
{
    <instrução1>
    .
    .
    .
    <instrução>
}
while (<condição>);
```

Estruturas de Controle de Fluxo

3. Laços de repetição (continuação)

Exemplo:

Dada uma reta $ax+by+c=0$ e cinco pontos, faça um programa para calcular, para cada ponto, o seguinte: se o ponto estiver no primeiro quadrante calcule e informe a distância do ponto a reta caso contrário escreva uma mensagem informando que o ponto não pertence ao primeiro quadrante.

Estruturas de Controle de Fluxo

```
#include <stdio.h>
#include <math.h>
main()
{
    float a,b,c,x,y;
    int contador=1;
    printf ("Equação da reta: ax+by+c=0\n");
    printf ("Coeficiente a da reta = ");
    scanf ("%f",&a);
    printf ("\nCoeficiente b da reta = ");
    scanf ("%f",&b);
    printf ("\nCoeficiente c da reta = ");
    scanf ("%f",&c);
```

```

do
{
    printf ("\nCoordenada x do ponto = ");
    scanf ("%f",&x);
    printf ("\nCoordenada y do ponto = ");
    scanf ("%f",&y);
    if (x>=0.0 && y>=0)
        printf ("\nA distancia do ponto a reta eh: %f",
            fabs(a*x+b*y+c)/sqrt(pow(a,2)+pow(b,2)));
    else
        printf ("\nO ponto nao esta no primeiro
quadrante!");
    contador++;
}
while (contador<=5);

```

Estruturas de Controle de Fluxo

3. Laços de repetição (continuação)

Exercício:

Faça um programa que recebe números naturais fornecidos pelo usuário, quando o usuário quiser parar a execução do programa, o mesmo fornecerá um número negativo. O programa deve retornar ao final o número de números naturais fornecidos pelo usuário (fazer dois programas utilizando cada uma das estruturas de repetição vistas).

Estruturas de Controle de Fluxo

3. Laços de repetição (continuação)

Exercício:

Faça um programa que receba um número inteiro positivo fornecido pelo usuário e retorne o seu fatorial na saída padrão. (fazer dois programas distintos utilizando cada um uma das estruturas de repetição vistas).