



# Algoritmos e Estrutura de Dados – I

# Unidade 5

## Linguagem C – Entrada de Dados

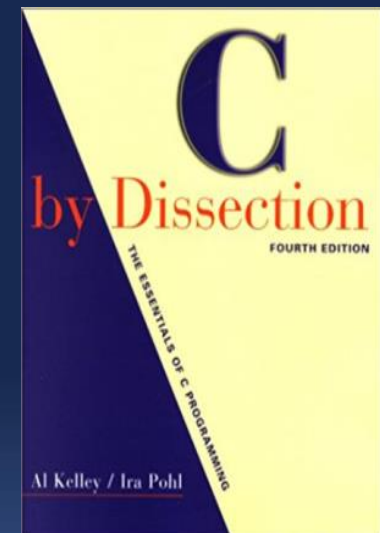
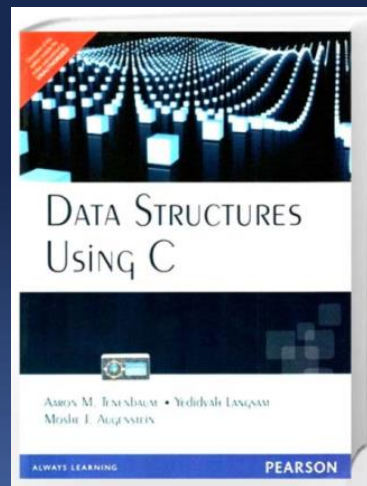
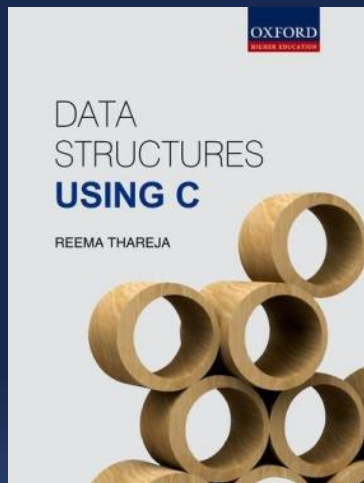


Prof. Aparecido V. de Freitas  
Doutor em Engenharia  
da Computação pela EPUSP  
aparecidovfreitas@gmail.com



# Bibliografia

- ✓ Data Structures using C - Oxford University Press - 2014
- ✓ Data Structures Using C - A. Tenenbaum, M. Augensem, Y. Langsam, Pearson 1995
- ✓ C By Dissection - Kelley, Pohl - Third Edition - Addison Wesley



# Função Scanf()

- ✓ Análoga à função `printf()`, mas usada para **entrada de dados**;
- ✓ Seu **primeiro argumento** é um **string** de **controle** que será usado para **formatar** as várias formas de interpretação dos caracteres de entrada;
- ✓ Após o **string de controle**, os demais argumentos são **endereços**;
- ✓ O símbolo **&** representa o **endereço** de um argumento (variável).

```
scanf(string-de-controle,lista-de-argumentos;
```

# Função Scanf() – Caracteres de Controle

- ✓ Quando o **teclado** é usado para entrada de dados em um programa, uma **sequência** de caracteres é **digitada** e essa **sequência** é **recebida** pelo programa;
- ✓ Esta sequência é chamada de **input stream**;
- ✓ Se "123" for digitado, quem digitou poderia imaginar os dados como um **decimal inteiro**, mas o programa os **recebe** como uma sequência de **caracteres**;
- ✓ A função **scanf()** pode ser usada para **converter** strings de dígitos decimais, tais como 123, para valores **inteiros** e armazená-los no **endereço** de memória apropriado.

## Função `scanf()` – Caracteres de Controle

Caractere de Conversão	Como os caracteres na entrada são convertidos
c	caractere
d	decimal Inteiro
f	ponto flutuante (float)
lf	ponto flutuante (double)
s	string

# Atenção

- ✓ Com `printf()`, o string de controle `"%f"` é usado tanto para `float` quanto para `double`;
- ✓ Com `scanf()`, o formato `"%f"` é usado para leitura de dados `float` e `"%lf"` é usado para leitura de dados `double`.

## Exercício 8 – Pgm\_08.C

- ✓ Escrever um programa que efetua leitura de um determinado **caractere** na entrada, armazene-o em alguma variável e **depois** o imprima na console. (ECO)

## Exercício 8 – Pgm\_08.C

```
#include <stdio.h>

int main ( ) {

    char entrada;

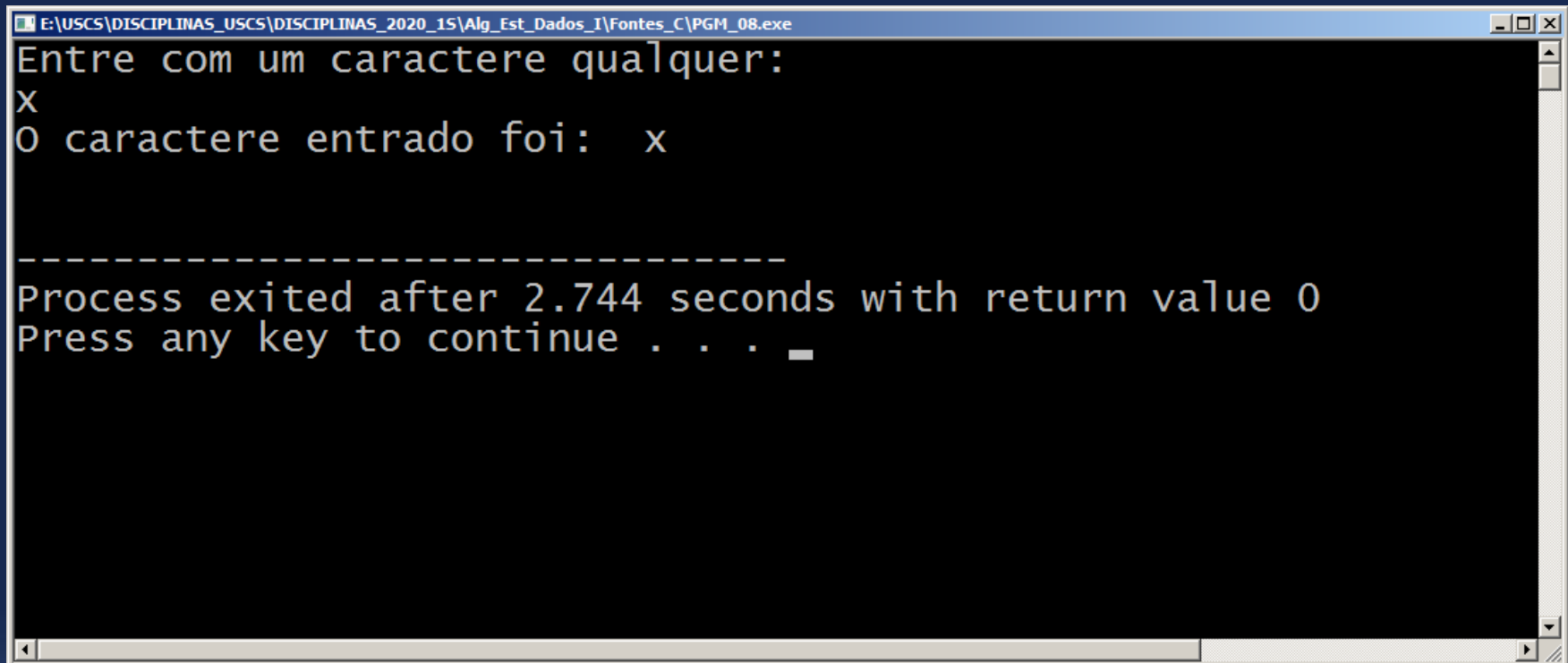
    printf("Entre com um caractere qualquer: \n");
    ↓
    scanf("%c",&entrada);

    printf ("0 caractere entrado foi:  %c\n\n",entrada);

    return 0;
}
```



# Exercício 8 – Pgm\_08.C



```
E:\USCS\DISCIPLINAS_USCS\DISCIPLINAS_2020_15\Alg_Est_Dados_I\Fontes_C\PGM_08.exe
Entre com um caractere qualquer:
x
O caractere entrado foi:  x

-----
Process exited after 2.744 seconds with return value 0
Press any key to continue . . . _
```

# Exercício 9 – Pgm\_09.C

- ✓ Escrever um programa que efetua leitura de um determinado valor **numérico inteiro** na entrada, armazene-o em alguma variável e depois o **imprima** na console.

# Exercício 9 – Pgm\_09.C

```
#include <stdio.h>
#include <locale.h>

int main ( ) {
    setlocale(LC_ALL, "Portuguese");
    ➡ int n;

    printf("Entre com um valor numérico inteiro: \n");
    scanf("%d",&n);
    printf ("O valor entrado foi: %d\n\n",n);

    return 0;
}
```

# Exercício 9 – Pgm\_09.C

```
E:\USCS\DISCIPLINAS_USCS\DISCIPLINAS_2020_15\Alg_Est_Dados_I\Fontes_C\PGM_09.exe
Entre com um valor numérico inteiro:
12345
O valor entrado foi: 12345

-----
Process exited after 3.399 seconds with return value 0
Press any key to continue . . . _
```

# Exercício 10 – Pgm\_10.C

- ✓ Escrever um programa que efetua leitura de um determinado valor **numérico decimal (float)** na entrada, armazene-o em alguma variável e depois o **imprima** na console.
- ✓ Obs. O valor deve ser impresso com **3** casas decimais.
- ✓ Dica: No **printf**, usar o string de controle: **"%.3f"**

# Exercício 10 – Pgm\_10.C

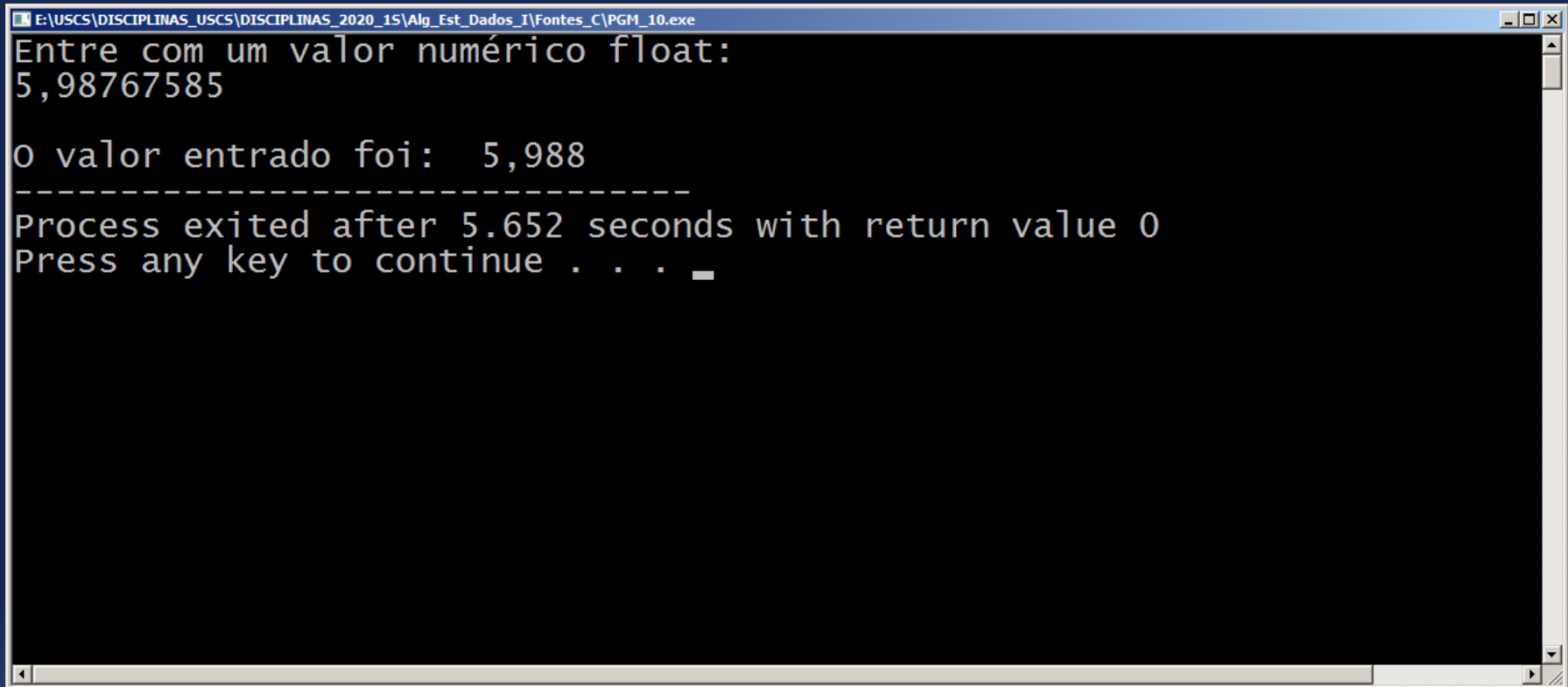
```
#include <stdio.h>
#include <locale.h>

int main ( ) {
    setlocale(LC_ALL, "Portuguese");
    ➡ float n;

    printf("Entre com um valor numérico float: \n");
    ↓
    scanf("%f",&n);

    printf ("\nO valor entrado foi: %.3f",n);
    ↓
    return 0;
}
```

# Exercício 10 – Pgm\_10.C



```
E:\USCS\DISCIPLINAS_USCS\DISCIPLINAS_2020_15\Alg_Est_Dados_I\Fontes_C\PGM_10.exe
Entre com um valor numérico float:
5,98767585

O valor entrado foi:  5,988
-----
Process exited after 5.652 seconds with return value 0
Press any key to continue . . .
```

# Exercício 11 – Pgm\_11.C

- ✓ Escrever um programa que efetua leitura de um determinado valor **numérico decimal (double)** na entrada, armazene-o em alguma variável e depois o **imprima** na console.
- ✓ Obs. O valor deve ser impresso com **2** casas decimais.
- ✓ Dica: No **printf**, usar o string de controle: **"%.2f"**



# Exercício 11 – Pgm\_11.C

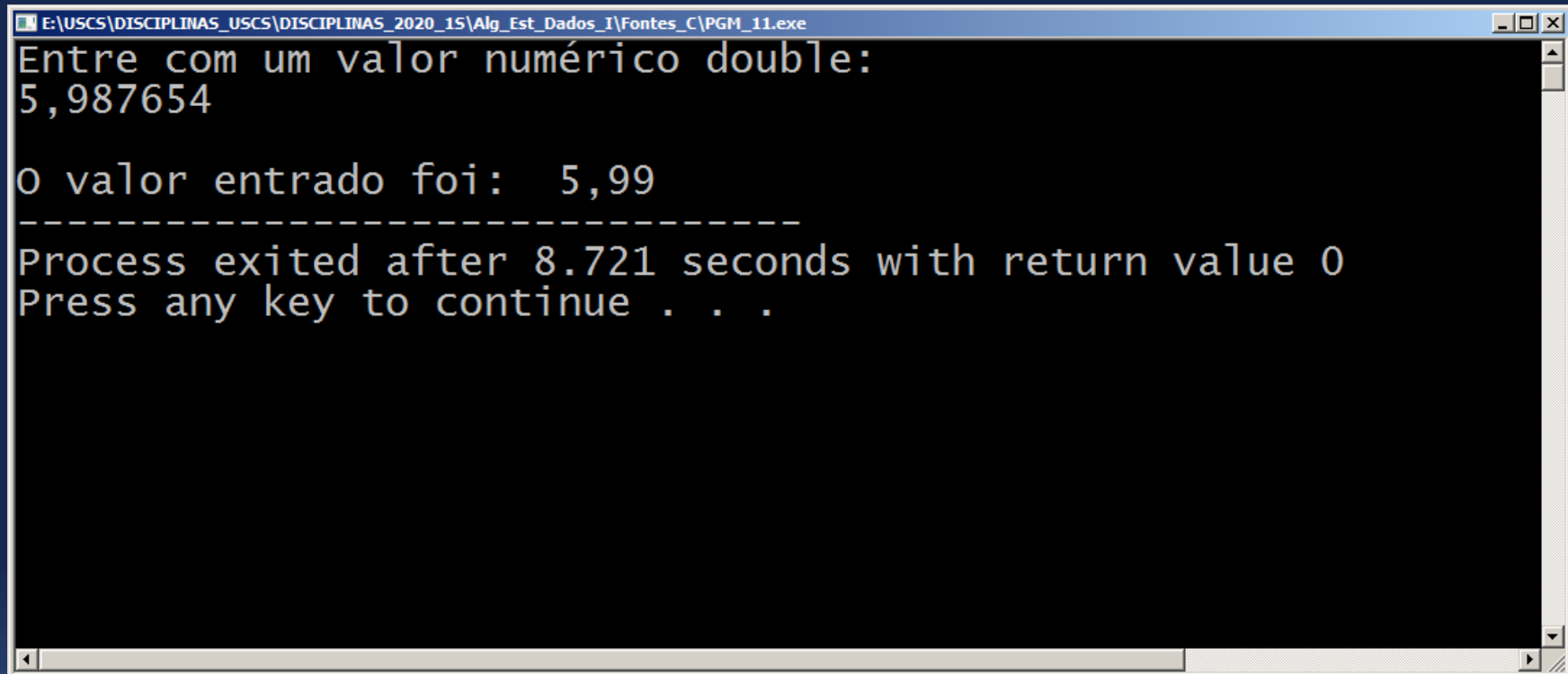
```
#include <stdio.h>
#include <locale.h>

int main ( ) {
    setlocale(LC_ALL, "Portuguese");
    ➡ double n;

    printf("Entre com um valor numérico double: \n");
    ↓
    scanf("%lf",&n);

    printf ("\nO valor entrado foi: %.2f",n);
    ↓
    return 0;
}
```

# Exercício 11 – Pgm\_11.C



```
E:\USCS\DISCIPLINAS_USCS\DISCIPLINAS_2020_15\Alg_Est_Dados_I\Fontes_C\PGM_11.exe
Entre com um valor numérico double:
5,987654

O valor entrado foi:  5,99
-----
Process exited after 8.721 seconds with return value 0
Press any key to continue . . .
```

## Exercício 12 – Pgm\_12.C

- ✓ Escrever um programa que efetua leitura do **primeiro nome** de uma pessoa, armazene-o em alguma variável e depois o **imprima** na console.

Dica: Declarar o texto de entrada como uma lista de caracteres:  
**char nome[40]; (Array de caracteres)**

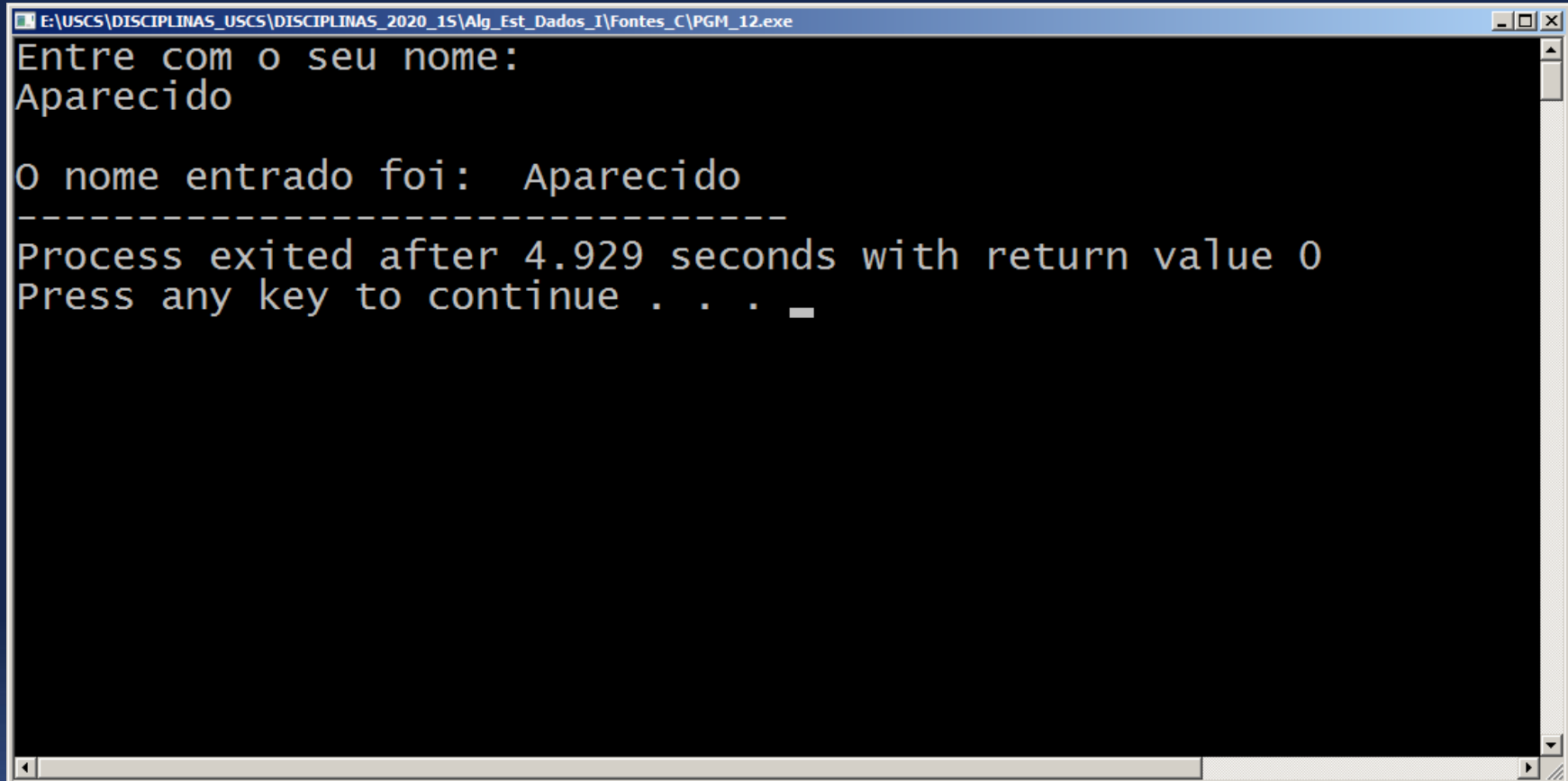
# Exercício 12 – Pgm\_12.C

```
#include <stdio.h>
#include <locale.h>

int main ( ) {
    setlocale(LC_ALL, "Portuguese");
    ➡ char nome[40];

    printf("Entre com o seu nome: \n");
    ↓
    scanf("%s",&nome);
    ↓
    printf ("\n0 nome entrado foi: %s",nome);
    ↓
    return 0;
}
```

# Exercício 12 – Pgm\_12.C



```
E:\USCS\DISCIPLINAS_USCS\DISCIPLINAS_2020_15\Alg_Est_Dados_I\Fontes_C\PGM_12.exe
Entre com o seu nome:
Aparecido

O nome entrado foi:  Aparecido
-----
Process exited after 4.929 seconds with return value 0
Press any key to continue . . . _
```

# Comentários

- `//` é usado para um simples comando;
- `/*` e `*/` é usado para comentar blocos de comandos

```
/* isto é um  
   comentário de  
       bloco */
```

```
// isto é um comentário de linha
```

# Standard Header Files

- **string.h** funções de manuseio de strings
- **stdlib.h** funções genéricas
- **stdio.h** funções para input/output
- **math.h** funções matemáticas
- **alloc.h** funções para alocação dinâmica de memória
- **conio.h** funções para limpeza de tela

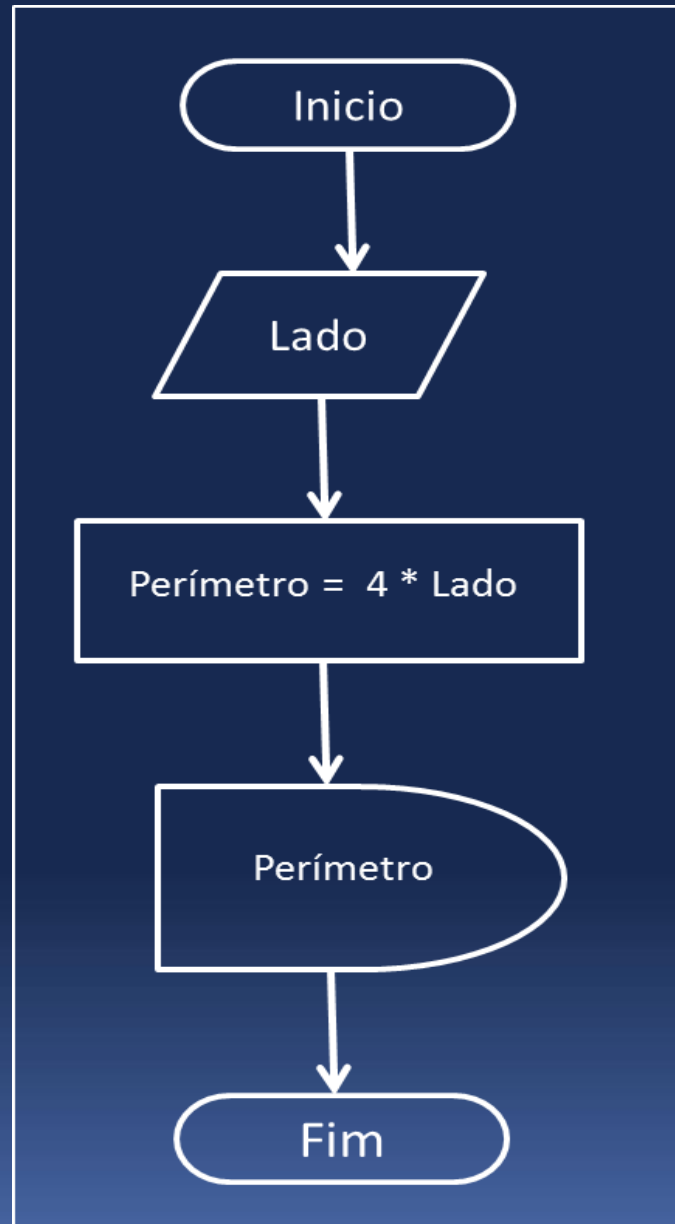
# Atividade 1

Codificar um programa com a linguagem C, para a **leitura** de um **valor numérico inteiro** que corresponde ao **lado** de um quadrado.

O programa deverá exibir em **tela** o resultado do **perímetro** do quadrado.



# Atividade 1



# Atividade 1

```
#include <stdio.h>
#include <locale.h>
int main() {

    setlocale(LC_ALL, "Portuguese");
    int lado, perimetro;

    printf("Entre com o lado do quadrado: \n") ;
    scanf("%d", &lado);

    perimetro = 4 * lado;

    printf("Perímetro = %d", perimetro);
    return 0;
}
```

# Atividade 1

```
E:\USCS\DISCIPLINAS_USCS\DISCIPLINAS_2020_15\Alg_Est_Dados_I\Fontes_C\Atividade_1.exe
Entre com o lado do quadrado:
10
Perímetro = 40
-----
Process exited after 86.99 seconds with return value 0
Press any key to continue . . . _
```

# Atividade 2

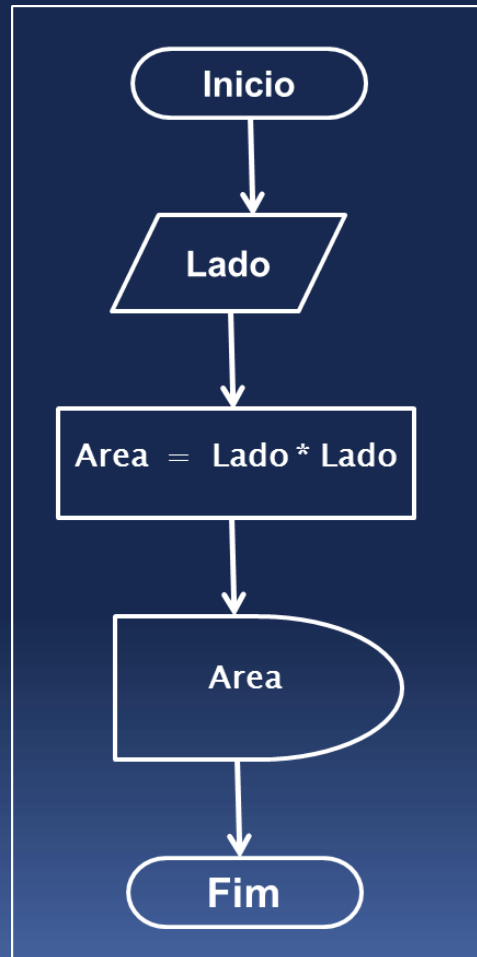
Codificar um programa com a linguagem C, para a **leitura** de um **valor numérico inteiro** que corresponde ao **lado** de um quadrado.

O programa deverá exibir em **tela** o resultado do **área** do quadrado.

# Atividade 2

Codificar um programa com a linguagem C, para a **leitura** de um **valor numérico inteiro** que corresponde ao **lado** de um quadrado.

O programa deverá exibir em **tela** o resultado do **área** do quadrado.



# Atividade 2



```
#include <stdio.h>
#include <locale.h>
int main() {

    setlocale(LC_ALL, "Portuguese");
    int lado, area;

    printf("Entre com o lado do quadrado: \n") ;
    scanf("%d", &lado);

    area = lado * lado;

    printf("Área = %d", area);
    return 0;
}
```



# Atividade 2



```
E:\USCS\DISCIPLINAS_USCS\DISCIPLINAS_2020_15\Alg_Est_Dados_I\Fontes_C\Atividade_2.exe
Entre com o lado do quadrado:
10
Area = 100
-----
Process exited after 3.382 seconds with return value 0
Press any key to continue . . .
```



# Atividade 3

Codificar um programa com a linguagem C, para a **leitura** de dois valores **numéricos inteiros** que correspondem aos **lados** de um retângulo.

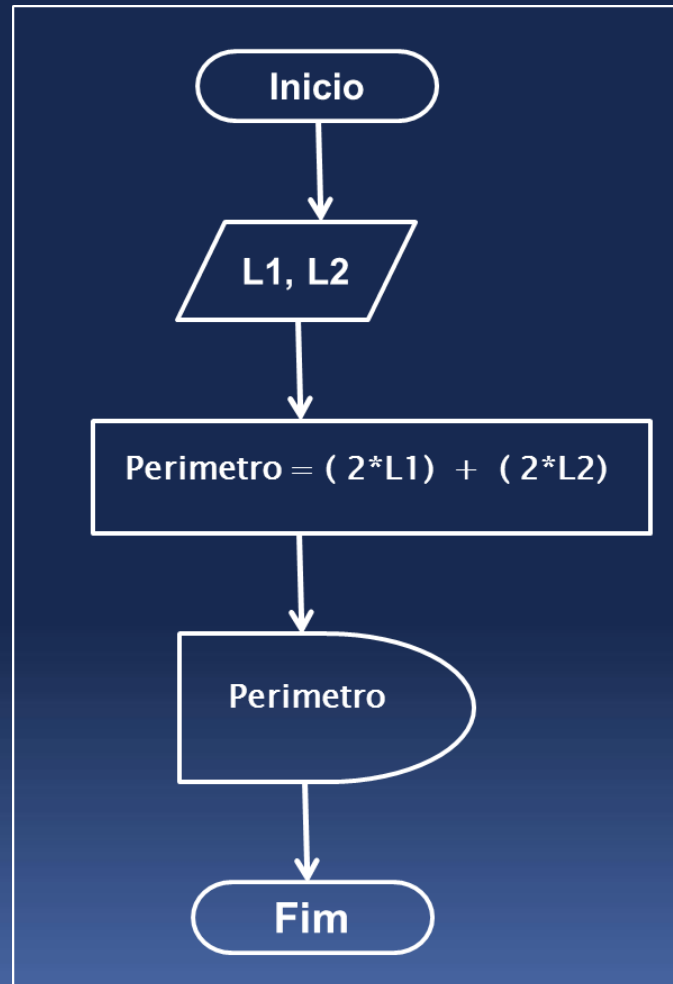
O programa deverá exibir em **tela** o resultado do **perímetro** do retângulo.



# Atividade 3

Codificar um programa com a linguagem C, para a **leitura** de dois valores **numéricos inteiros** que correspondem aos **lados** de um retângulo.

O programa deverá exibir em **tela** o resultado do **perímetro** do retângulo.



# Atividade 3

```
#include <stdio.h>
#include <locale.h>
int main() {

    setlocale(LC_ALL, "Portuguese");
    int L1, L2, perimetro;

    printf("Entre com o primeiro lado do retângulo: \n") ;
    scanf("%d", &L1);

    printf("Entre com o segundo lado do retângulo: \n") ;
    scanf("%d", &L2);
    perimetro = (2 * L1) + (2 * L2);

    printf("Perímetro = %d", perimetro);
    return 0;
}
```

# Atividade 3

```
E:\USCS\DISCIPLINAS_USCS\DISCIPLINAS_2020_15\Alg_Est_Dados_I\Fontes_C\Atividade_3.exe
Entre com o primeiro lado do retângulo:
10
Entre com o segundo lado do retângulo:
20
Perímetro = 60
-----
Process exited after 4.908 seconds with return value 0
Press any key to continue . . .
```

# Atividade 4

Codificar um programa com a linguagem C, para a **leitura** de dois valores **numéricos inteiros** que correspondem aos **lados** de um retângulo.

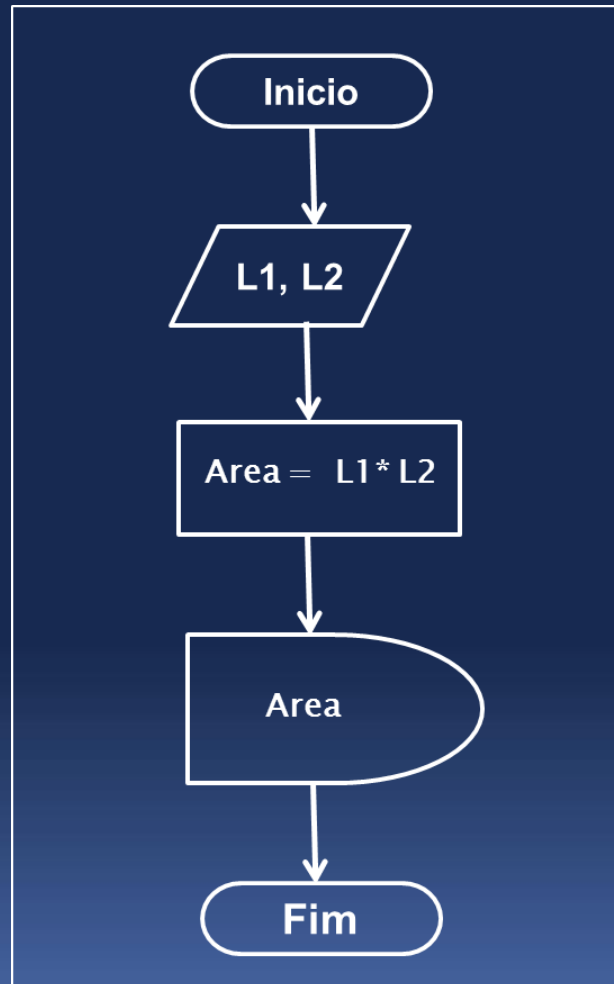
O programa deverá exibir em **tela** o resultado do **área** do retângulo.

# Atividade 4



Codificar um programa com a linguagem C, para a **leitura** de dois valores **numéricos inteiros** que correspondem aos **lados** de um retângulo.

O programa deverá exibir em **tela** o resultado do **área** do retângulo.



# Atividade 4



```
#include <stdio.h>
#include <locale.h>
int main() {

    setlocale(LC_ALL, "Portuguese");
    int L1, L2, area;

    printf("Entre com o primeiro lado do retângulo: \n") ;
    scanf("%d", &L1);

    printf("Entre com o segundo lado do retângulo: \n") ;
    scanf("%d", &L2);
    area = L1 * L2;

    printf("Área = %d", area);
    return 0;
}
```



# Atividade 4

```
E:\USCS\DISCIPLINAS_USCS\DISCIPLINAS_2020_15\Alg_Est_Dados_I\Fontes_C\Atividade_4.exe
Entre com o primeiro lado do retângulo:
10
Entre com o segundo lado do retângulo:
20
Area = 200
-----
Process exited after 7.238 seconds with return value 0
Press any key to continue . . .
```

# Atividade 5

Codificar um programa com a linguagem C, para a **leitura** de um valor **numérico inteiro** que corresponde ao **lado** de um quadrado.

Caso o número seja **ímpar**, o algoritmo deverá enviar a mensagem “**Valor Inválido**” e **encerrar**. Caso contrário, o programa deverá exibir em **tela** o resultado do **área** do quadrado.

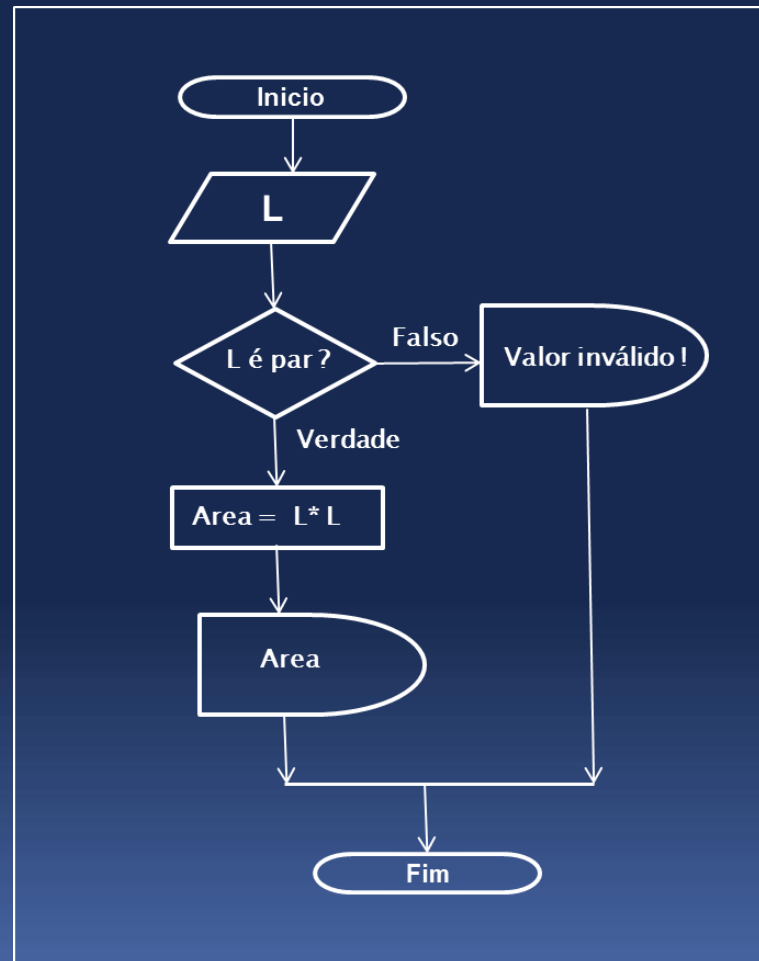


# Atividade 5



Codificar um programa com a linguagem C, para a **leitura** de um valor **numérico inteiro** que corresponde ao **lado** de um quadrado.

Caso o número **não** seja **par**, o algoritmo deverá enviar a mensagem “**Valor Inválido**” e encerrar. Caso contrário, o programa deverá exibir em **tela** o resultado do **área** do quadrado.



# Atividade 5



```
#include <stdio.h>
#include <locale.h>
int main() {

    setlocale(LC_ALL, "Portuguese");
    int L, area;

    printf("Entre com o lado do quadrado: \n") ;
    scanf("%d", &L);

    if (L % 2 == 1 ) {
        printf("Valor Inválido...");
        return 1;
    }

    area = L * L;

    printf("Área = %d", area);
    return 0;
}
```



# Atividade 5



```
E:\USCS\DISCIPLINAS_USCS\DISCIPLINAS_2020_15\Alg_Est_Dados_I\Fontes_C\Atividade_5.exe
Entre com o lado do quadrado:
10
Area = 100
-----
Process exited after 3.075 seconds with return value 0
Press any key to continue . . . _
```

```
E:\USCS\DISCIPLINAS_USCS\DISCIPLINAS_2020_15\Alg_Est_Dados_I\Fontes_C\Atividade_5.exe
Entre com o lado do quadrado:
9
Valor Inválido...
-----
Process exited after 2.674 seconds with return value 1
Press any key to continue . . . _
```

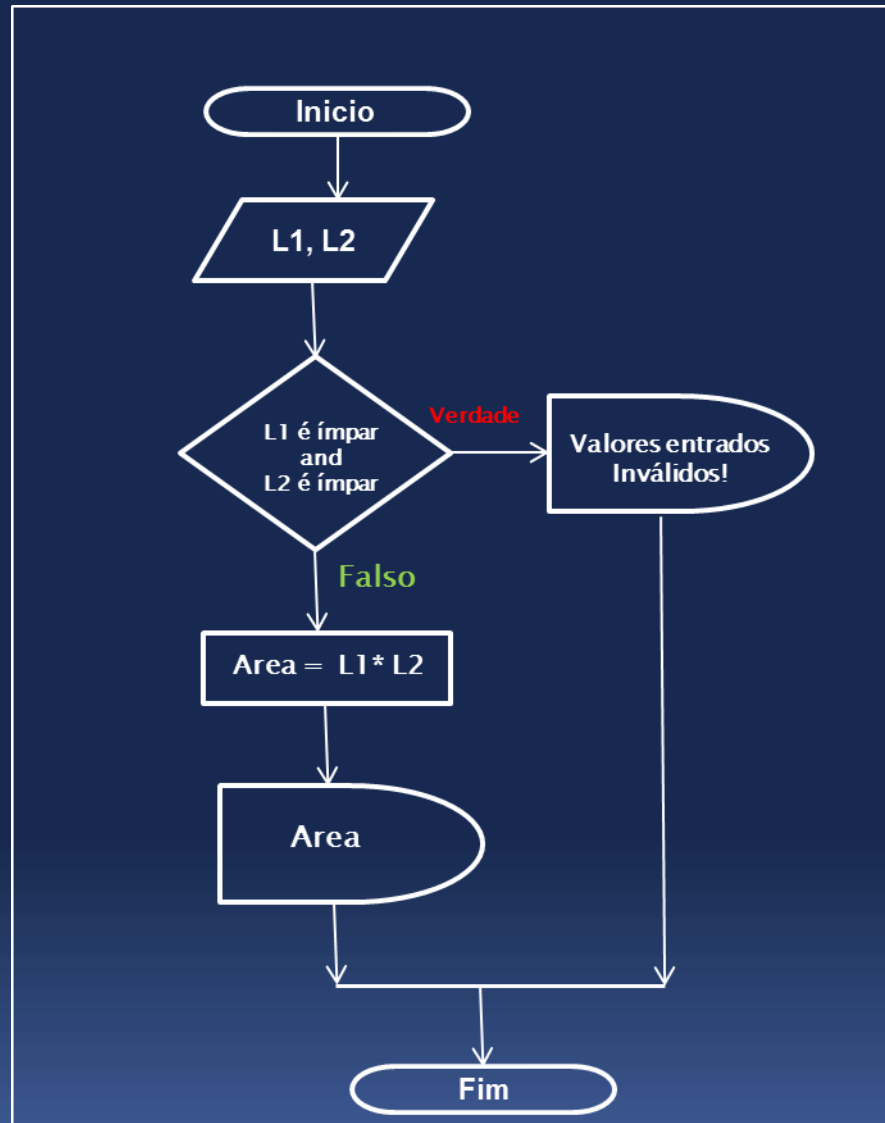


# Atividade 6

Codificar um programa com a linguagem C, para a **leitura** de dois valores **numéricos inteiros** que correspondem aos **lados** de um retângulo.

Caso os **dois** valores entrados sejam **ímpares**, o programa deverá enviar a mensagem “**Valores entrados Inválidos**” e **encerrar**. Caso contrário, o programa deverá exibir em **tela** o resultado do **área** do retângulo.

# Atividade 6



# Atividade 6



```
#include <stdio.h>
#include <locale.h>
int main() {

    setlocale(LC_ALL, "Portuguese");
    int L1, L2, area;

    printf("Entre com o primeiro lado do retângulo: \n") ;
    scanf("%d", &L1);

    printf("Entre com o segundo lado do retângulo: \n") ;
    scanf("%d", &L2);

    if ( (L1 % 2 == 1) && (L2 % 2 == 1) ) {
        printf("Valores entrados Inválidos...");
        return 1;
    }

    area = L1 * L2;

    printf("Área = %d", area);
    return 0;
}
```



# Atividade 6



```
E:\USCS\DISCIPLINAS_USCS\DISCIPLINAS_2020_15\Alg_Est_Dados_I\Fontes_C\Atividade_6.exe
Entre com o primeiro lado do retângulo:
2
Entre com o segundo lado do retângulo:
4
Area = 8
-----
Process exited after 7.736 seconds with return value 0
Press any key to continue . . .
```

```
E:\USCS\DISCIPLINAS_USCS\DISCIPLINAS_2020_15\Alg_Est_Dados_I\Fontes_C\Atividade_6.exe
Entre com o primeiro lado do retângulo:
2
Entre com o segundo lado do retângulo:
3
Area = 6
-----
Process exited after 4.891 seconds with return value 0
Press any key to continue . . .
```



# Atividade 6



```
E:\USCS\DISCIPLINAS_USCS\DISCIPLINAS_2020_15\Alg_Est_Dados_I\Fontes_C\Atividade_6.exe
Entre com o primeiro lado do retângulo:
3
Entre com o segundo lado do retângulo:
2
Area = 6
-----
Process exited after 4.127 seconds with return value 0
Press any key to continue . . .
```

```
E:\USCS\DISCIPLINAS_USCS\DISCIPLINAS_2020_15\Alg_Est_Dados_I\Fontes_C\Atividade_6.exe
Entre com o primeiro lado do retângulo:
3
Entre com o segundo lado do retângulo:
5
Valores entrados Inválidos...
-----
Process exited after 3.9 seconds with return value 1
Press any key to continue . . .
```





# Atividade 7

Codificar um programa com a linguagem C, para a **leitura** de um valor **numérico inteiro**.

Caso o número entrado seja **menor ou igual a zero**, o programa deverá enviar a mensagem “**Valor entrado Inválido**” e encerrar. Caso contrário, o programa deverá exibir em **tela** o resultado do **dobro** do número entrado.

# Atividade 7

```
#include <stdio.h>
#include <locale.h>
int main() {

    setlocale(LC_ALL, "Portuguese");
    int valor, dobro;

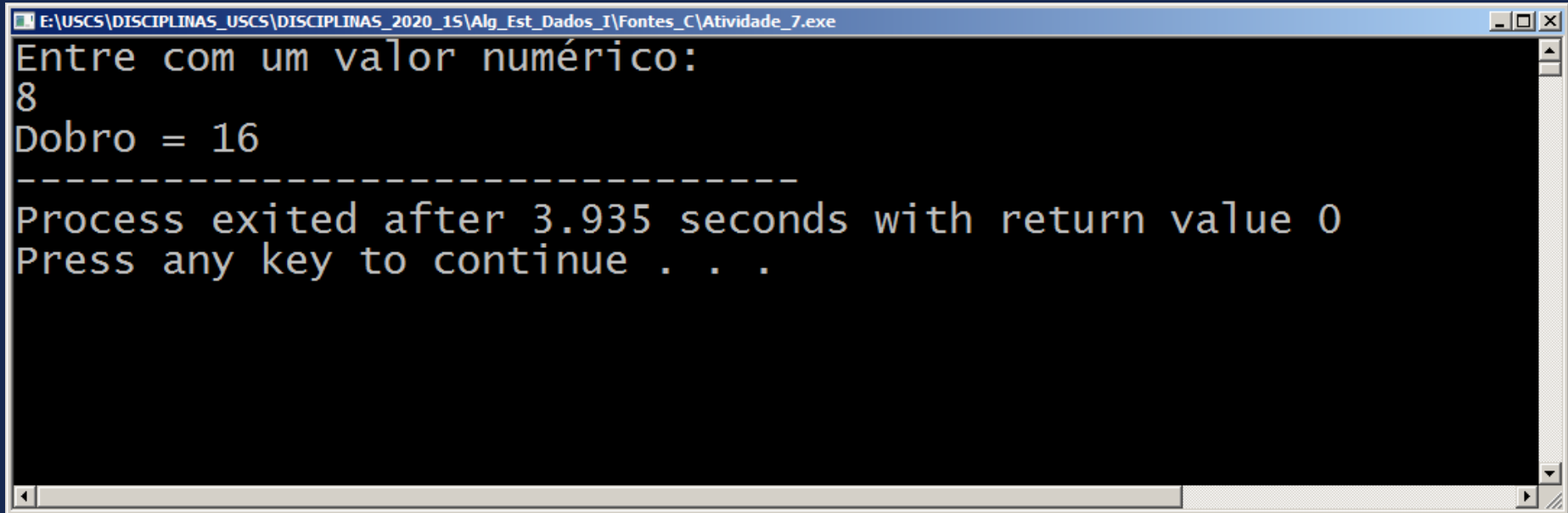
    printf("Entre com um valor numérico: \n") ;
    scanf("%d", &valor);

    if ( valor <= 0 ) {
        printf("Valor entrado Inválido...");
        return 1;
    }

    dobro = 2 * valor;

    printf("Dobro = %d", dobro);
    return 0;
}
```

# Atividade 7



```
E:\USCS\DISCIPLINAS_USCS\DISCIPLINAS_2020_15\Alg_Est_Dados_I\Fontes_C\Atividade_7.exe
Entre com um valor numérico:
8
Dobro = 16
-----
Process exited after 3.935 seconds with return value 0
Press any key to continue . . .
```

# Atividade 8

Codificar um programa com a linguagem C, para a **leitura** de dois valores **numéricos inteiros**.

Caso os **menores ou iguais a zero**, o programa deverá enviar a mensagem “**Valores entrados Inválidos**” e encerrar. Caso contrário, o programa deverá exibir em **tela** o resultado da **soma** dos dois valores entrados.

# Atividade 8

```
#include <stdio.h>
#include <locale.h>
int main() {

    setlocale(LC_ALL, "Portuguese");
    int valor1, valor2, soma;

    printf("Entre com o primeiro valor numérico: \n") ;
    scanf("%d", &valor1);

    printf("Entre com o segundo valor numérico: \n") ;
    scanf("%d", &valor2);

    if ( ( valor1 <= 0 ) && (valor2 <= 0 ) ) {
        printf("Valores entrados Inválidos...");
        return 1;
    }

    soma = valor1 + valor2;

    printf("Soma = %d", soma);
    return 0;
}
```

# Atividade 8

```
E:\USCS\DISCIPLINAS_USCS\DISCIPLINAS_2020_15\Alg_Est_Dados_I\Fontes_C\Atividade_8.exe
Entre com o primeiro valor numérico:
1
Entre com o segundo valor numérico:
55
Soma = 56
-----
Process exited after 4.898 seconds with return value 0
Press any key to continue . . . _
```

# Atividade 9

Codificar um programa com a linguagem C, para a **leitura** de dois valores **numéricos inteiros**.

Caso os **dois valores entrados** sejam **menores ou iguais a zero**, o programa deverá enviar a mensagem “**Valores entrados Inválidos**” e encerrar. Caso contrário, o programa deverá exibir em **tela** o **maior** dos valores entrados.

# Atividade 9



```
#include <stdio.h>
#include <locale.h>
int main() {

    setlocale(LC_ALL, "Portuguese");
    int valor1, valor2, maior;

    printf("Entre com o primeiro valor numérico: \n") ;
    scanf("%d", &valor1);

    printf("Entre com o segundo valor numérico: \n") ;
    scanf("%d", &valor2);

    if ( ( valor1 <= 0 ) && (valor2 <= 0 ) ) {
        printf("Valores entrados Inválidos...");
        return 1;
    }

    if (valor1 >= valor2 )
        maior = valor1;
    else maior = valor2;

    printf("Maior = %d", maior);
    return 0;
}
```





# Atividade 9

```
E:\USCS\DISCIPLINAS_USCS\DISCIPLINAS_2020_15\Alg_Est_Dados_I\Fontes_C\Atividade_9.exe
Entre com o primeiro valor numérico:
5
Entre com o segundo valor numérico:
99
Maior = 99
-----
Process exited after 3.229 seconds with return value 0
Press any key to continue . . . _
```

# Atividade 10

Codificar um programa com a linguagem C, para a **leitura** de dois valores **numéricos inteiros**.

Caso os **dois valores entrados** sejam **menores ou iguais a zero**, o programa deverá enviar a mensagem “**Valores entrados Inválidos**” e encerrar. Caso os **valores entrados sejam iguais**, o programa deverá exibir em tela a mensagem “**Valores entrados são iguais**”. Caso contrário, o programa deverá exibir em **tela** o **maior** dos valores entrados.

# Atividade 10



```
#include <stdio.h>
#include <locale.h>
int main() {

    setlocale(LC_ALL, "Portuguese");
    int valor1, valor2, maior;

    printf("Entre com o primeiro valor numérico: \n") ;
    scanf("%d", &valor1);

    printf("Entre com o segundo valor numérico: \n") ;
    scanf("%d", &valor2);

    if ( ( valor1 <= 0 ) && (valor2 <= 0 ) ) {
        printf("Valores entrados Inválidos...");
        return 1;
    }

    if (valor1 == valor2) {
        printf("Valores entrados são iguais...");
    }
    else {
        if (valor1 > valor2 ) {
            maior = valor1;
        }
        else {
            maior = valor2;
        }
        printf ("Maior valor:  %d ", maior);
    }
    return 0;
}
```



# Atividade 10

```
E:\USCS\DISCIPLINAS_USCS\DISCIPLINAS_2020_15\Alg_Est_Dados_1\Fontes_C\Atividade_10.exe
Entre com o primeiro valor numérico:
7
Entre com o segundo valor numérico:
6
Maior valor:  7
-----
Process exited after 1.856 seconds with return value 0
Press any key to continue . . .
```

# O comando while

- ✓ Comandos em um programa escrito em C são normalmente executados uns após outros;
- ✓ Esse procedimento é conhecido por **Fluxo Sequencial de Execução**;
- ✓ Ações **repetitivas** de comandos podem ser feitas pelo comando **while**;

# O comando while

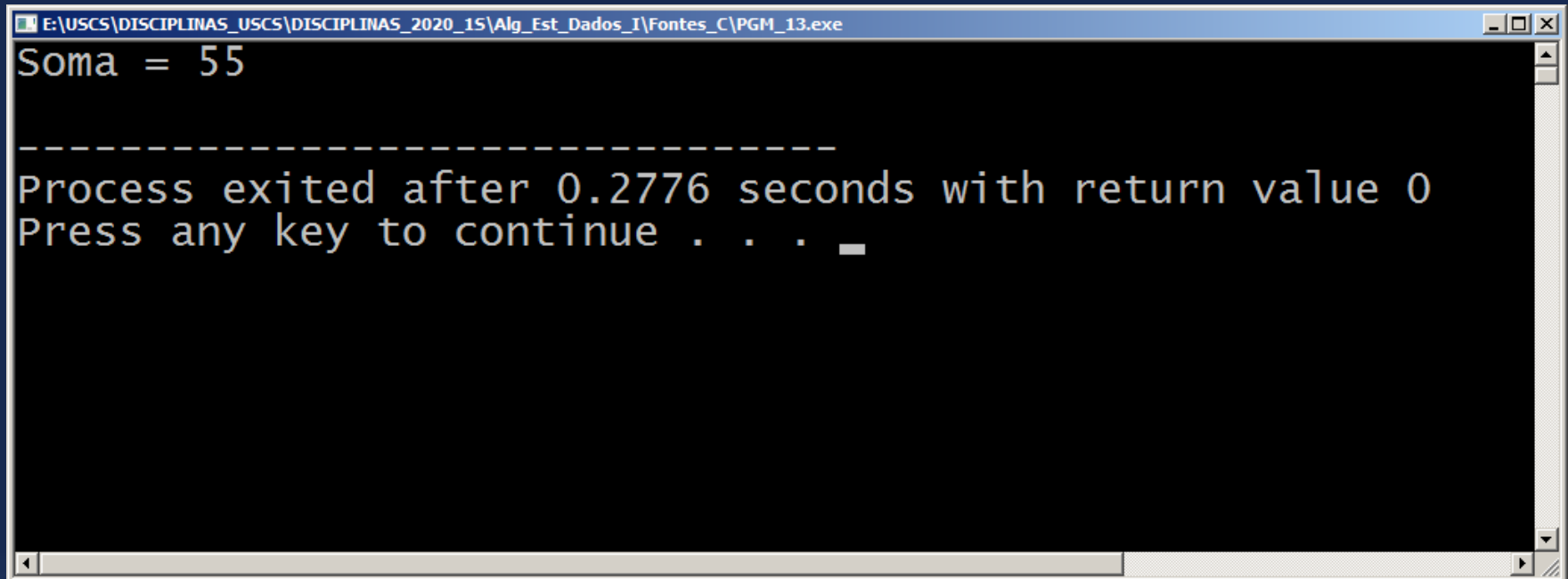
```
#include <stdio.h>
#include <locale.h>

int main ( ) {
    setlocale(LC_ALL, "Portuguese");
    int i = 1, soma = 0;

    while (i <= 10) {
        soma = soma + i;
        i = i + 1;
    }
    printf("Soma = %d\n", soma);

    return 0;
}
```

# O comando while



```
E:\USCS\DISCIPLINAS_USCS\DISCIPLINAS_2020_15\Alg_Est_Dados_I\Fontes_C\PGM_13.exe
Soma = 55
-----
Process exited after 0.2776 seconds with return value 0
Press any key to continue . . . _
```

# scanf() – observação

- ✓ Como vimos no início desta unidade, a função **scanf()** efetua leitura de informações para o programa;
- ✓ Adicionalmente, **scanf()** efetua conversões de dados o que é materializado pela string de conversão %.
- ✓ Por exemplo, %d significa que o dado entrado pelo usuário será convertido para um valor numérico decimal inteiro;
- ✓ A função **scanf()** **retorna** um valor que corresponde ao resultado da operação efetuada. Se a conversão foi feita com **sucesso**, **scanf()** **retorna 1**, caso contrário **retorna -1**;



# Exemplo – Retorno scanf()

```
#include <stdio.h>
#include <locale.h>
#define true 1

int main ( ) {
    setlocale(LC_ALL, "Portuguese");
    int valor, retorno, soma = 0;

    printf("\nEntre com um valor qualquer: ");
    while (true) {

        retorno = scanf("%d",&valor);
        if (retorno == true)
            soma = soma + valor;
        else
            break;
        printf("\nEntre com outro valor qualquer: ");
    }
    printf("Soma = %d\n", soma);

    return 0;
}
```

# Exemplo – Retorno scanf()



```
E:\USCS\DISCIPLINAS_USCS\DISCIPLINAS_2020_15\Alg_Est_Dados_I\Fontes_C\PGM_13.exe
Entre com um valor qualquer: 1
Entre com outro valor qualquer: 2
Entre com outro valor qualquer: 3
Entre com outro valor qualquer: 4
Entre com outro valor qualquer: 5
Entre com outro valor qualquer: x
Soma = 15

-----
Process exited after 6.41 seconds with return value 0
Press any key to continue . . .
```



# Elementos Léxicos

- ✓ Um programa em C pode ter os seguintes caracteres:

Characters that can be used in a program									
lowercase letters	a	b	c	...	z				
uppercase letters	A	B	C	...	Z				
digits	0	1	2	3	4	5	6	7	8 9
other characters	+ - * / = ( ) { } [ ] < > ' " ! @ # \$ % & _   ^ ~ \ . , ; : ?								
white space characters	<i>blank, newline, tab, etc.</i>								

# Operadores ++ e --

- ✓ São operadores unários que incrementam ou decrementam de 1 unidade uma variável.

```
#include <stdio.h>
#include <locale.h>
#define true 1

int main ( ) {
    setlocale(LC_ALL, "Portuguese");
    int A, B;

    printf("\nEntre com um valor numérico: ");
    scanf("%d",&A);

    printf("\nEntre com um outro valor numérico: ");
    scanf("%d",&B);

    A++;
    B++;

    printf("\nSomando uma unidade em cada variável...\n");

    printf("A = %d\n", A);
    printf("B = %d\n", B);

    return 0;
}
```

# Operadores ++ e --

```
E:\USCS\DISCIPLINAS_USCS\DISCIPLINAS_2020_15\Alg_Est_Dados_I\Fontes_C\PGM_14.exe

Entre com um valor numérico: 55
Entre com um outro valor numérico: 88
Somando uma unidade em cada variável...
A = 56
B = 89

-----
Process exited after 4.778 seconds with return value 0
Press any key to continue . . .
```

# Operadores ++ e --

```
#include <stdio.h>
#include <locale.h>
#define true 1

int main ( ) {
    setlocale(LC_ALL, "Portuguese");
    int A, B;

    printf("\nEntre com um valor numérico: ");
    scanf("%d",&A);

    printf("\nEntre com um outro valor numérico: ");
    scanf("%d",&B);

    ++A;
    ++B;

    printf("\nSomando uma unidade em cada variável...\n");

    printf("A = %d\n", A);
    printf("B = %d\n", B);

    return 0;
}
```

# Operadores ++ e --

```
E:\USCS\DISCIPLINAS_USCS\DISCIPLINAS_2020_15\Alg_Est_Dados_I\Fontes_C\PGM_14.exe

Entre com um valor numérico: 33
Entre com um outro valor numérico: 11

Somando uma unidade em cada variável...
A = 34
B = 12

-----
Process exited after 4.427 seconds with return value 0
Press any key to continue . . .
```

# Operador Atribuição

✓ Definidos por:  $+=$  ,  $-=$  ,  $*=$  ,  $/=$

$A += 2;$        $\Leftrightarrow$        $A = A + 2;$

$B -= 5;$        $\Leftrightarrow$        $B = B - 5;$

$C *= 3;$        $\Leftrightarrow$        $C = C * 3;$

$D /= 2;$        $\Leftrightarrow$        $D = D / 2;$



# Operador Atribuição

```
#include <stdio.h>
#include <locale.h>

int main ( ) {
    setlocale(LC_ALL, "Portuguese");

    int A, B, C, D;

    A = 1;
    B = 20;
    C = 3;
    D = 9;

    A += 2;
    B -= 5;
    C *= 3;
    D /= 5;

    printf("A = %d\n", A);
    printf("B = %d\n", B);
    printf("C = %d\n", C);
    printf("D = %d\n", D);

    return 0;
}
```

# Operador Atribuição

```
E:\USCS\DISCIPLINAS_USCS\DISCIPLINAS_2020_15\Alg_Est_Dados_I\Fontes_C\PGM_14.exe
A = 3
B = 15
C = 9
D = 1

-----
Process exited after 0.2916 seconds with return value 0
Press any key to continue . . .
```

# Sistema C

- ✓ Composto por pré-compilador, compilador, bibliotecas e utilitários;
- ✓ Segue um exemplo de um programa que usa stdlib.h

```
#include <stdio.h>
#include <stdlib.h>
#include <locale.h>

int main ( ) {
    setlocale(LC_ALL, "Portuguese");

    int i, n;

    printf("Este programa exibe alguns números inteiros aleatórios...\n\n");
    printf("Entre com a quantidade de números a serem exibidos: ") ;

    scanf("%d", &n);

    for (i=0; i< n; i++) {

        if (i%6 == 0)
            printf ("\n");

        printf ("%10d", rand() );

    }

    return 0;
}
```

# Sistema C



```
E:\USCS\DISCIPLINAS_USCS\DISCIPLINAS_2020_15\Alg_Est_Dados_I\Fontes_C\PGM_14.exe
29292      5997      17549      29556      25561      31627
 6467      29541      26129      31240      27813      29174
20601      6077      20215      8683      8213      23992
25824      5601      23392      15759      2670      26428
28027      4084      10075      18786      15498      24970
 6287      23847      32604      503      21221      22663
 5706      2363      9010      22171      27489      18240
12164      25542      7619      20913      7591      6704
31818      9232      750      25205      4975      1539
 303      11422      21098      11247      13584      13648
 2971      17864      22913      11075      21545      28712
17546      18678      1769      15262      8519      13985
28289      15944      2865      18540      23245      25508
28318      27870      9601      28323      21132      24472
27152      25087      28570      29763      29901      17103
14423      3527      11600      26969      14015      5565
 28      21543      25347      2088      2943      12637
22409      26463      5049      4681      1588      11342
 608      32060      21221      1758      29954      20888
14146      690      7949      12843      21430      25620
 748      27067

-----
Process exited after 3.973 seconds with return value 0
Press any key to continue . . .
```

