

# Respostas dos Exercícios

## Programando em Linguagem C e C++

Com o Microsoft Visual Studio

Antônio Henrique Reis



**viena**

1ª Edição  
Santa Cruz do Rio Pardo/SP  
Editora Viena  
2015

---

## Capítulo 1

---

1.

Devemos colocar o comando: `system("PAUSE");` no final do programa, antes do comando: `return 0;`

Isto faz o programa fazer uma pausa e apresentar na tela uma mensagem: Pressione qualquer tecla para continuar. . .

2.

Devemos compilar o programa, que é o processo onde será verificada a sintaxe do código fonte escrito. Para compilar o código fonte no VS devemos acessar o menu BUILD e, em seguida, clicar em Build Solution (F7). Se não ocorreram erros de sintaxe podemos testar a execução do programa através do menu DEBUG, clicando no comando Start Debugging (F5).

3.

Encontramos os seguintes erros e escrevemos a linha de forma correta:

```
printf(Escrevendo uma mensagem na tela")
```

A mensagem: Escrevendo uma mensagem na tela deveria ter sido inserida entre as "aspas".

No final da linha faltou o ";".

A linha deveria ter sido escrito da seguinte forma:

```
printf("Escrevendo uma mensagem na tela");
```

Na penúltima linha:

```
sistem("Pause");
```

O comando: "sistem" está escrito errado, deveria ter sido escrito da seguinte forma: "system".

A palavra Pause deveria estar entre aspas.

Portanto, esta linha deveria ter sido escrito da seguinte forma:

```
system("Pause");
```

## 4.

```
// Capitulo001_Exercicio04.cpp : Defines the entry point for the
console application.
//

#include "stdafx.h"
#include "stdlib.h"

int _tmain(int argc, _TCHAR* argv[])
{
    printf("Meu nome: João Silva \n");
    printf("Minha mae: Maria Silva \n");
    printf("Minha vo: Setembrina Silva \n");

    system("Pause");
    return 0;
}
```

## 5.

```
// Capitulo001_Exercicio05.cpp : Defines the entry point for the
console application.
//

#include "stdafx.h"
#include "stdlib.h"

int _tmain(int argc, _TCHAR* argv[])
{
    printf("MEU \n");
    printf("\tSEGUNDO \n");
    printf("\t\tPROGRAMA \n");
    printf("\t\t\tTEM LINGUAGEM C\n");

    system("PAUSE");
    return 0;
}
```

## 4 • RESPOSTAS DOS EXERCÍCIOS

### 6.

```
// Capitulo001_Exercicio06.cpp : Defines the entry point for the
console application.
//

#include "stdafx.h"
#include "stdlib.h"

int _tmain(int argc, _TCHAR* argv[])
{
    printf("\\\\\\"Terceiro programa\\\\\\" \n");

    system("PAUSE");
    return 0;
}
```

### 7.

```
// Capitulo001_Exercicio07.cpp : Defines the entry point for the
console application.
//

#include "stdafx.h"
#include "stdlib.h"

int _tmain(int argc, _TCHAR* argv[])
{
    printf("Meu \\\"querido amigo\\\" mudou-se! \n");

    system("PAUSE");
    return 0;
}
```

**8.**

```
// Capitulo001_Exercicio08.cpp : Defines the entry point for the
console application.
//

#include "stdafx.h"
#include "stdlib.h"

int _tmain(int argc, _TCHAR* argv[])
{
    printf("Primeira linha \n\n");
    printf("Terceira linha \n\n");
    printf("Quinta linha \n\n\n");
    printf("Oitava linha \n");

    system("PAUSE");
    return 0;
}
```

---

## Capítulo 2

---

**1.**

Case Sensitive, significa que a linguagem diferencia as letras maiúsculas das letras minúsculas em seus comandos. Em linguagem C e C++ devemos escrever os comandos em letras minúsculas.

**2.**

Sistema de numeração Binário:

Significa que iremos trabalhar com números na base 2, ou seja, com apenas dois dígitos numéricos. Na programação usaremos apenas o número "0" e "1".

O número "3" em binário é: "11".

O número "10" em binário é: "1010".

O número "15" em binário é: "1111".

O número "20" em binário é: "10100".

Sistema de numeração Hexadecimal:

Significa que iremos trabalhar com números na base 16, serão utilizados 16 dígitos.

Hexadecimal={0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}

O número "3" em Hexadecimal é: "3".

O número "10" em Hexadecimal é: "A".

## 6 • RESPOSTAS DOS EXERCÍCIOS

O número “15” em Hexadecimal é: “F”.  
O número “20” em Hexadecimal é: “14”.

Sistema de numeração Octal:

Significa que iremos trabalhar com números na base 8, serão utilizados 8 dígitos.

Octal={0, 1, 2, 3, 4, 5, 6, 7, 8}

O número “3” em Octal é: “3”.

O número “10” em Octal é: “12”.

O número “15” em Octal é: “17”.

O número “20” em Octal é: “24”.

### 3.

Escrevendo e compilando o programa, apesar de não ter sido utilizada nenhuma indentação e escrito em uma única linha o programa não tem erros e funciona, seu resultado será a apresentação na tela da mensagem: “Meu primeiro programa”

### 4.

Neste programa, quando compilarmos, será apresentada uma mensagem de erro, pois a chamada da função: PRINTF está escrita em maiúsculo e deveria ter sido escrita em minúsculo: printf.

### 5.

Neste programa teremos os seguintes resultados:

```
Fiz 2 programas
Letra C apresentada
Frase escrita
Numero: 34
O aumento foi de 10%
```

### 6.

Neste exercício veremos que os comentários não são lidos pelo compilador do programa e não fazem diferença alguma no código executável do programa.

### 7.

Neste exercício o compilador vai identificar um erro na linha:

Na primeira linha não colocamos as aspas no final da frase.

Na segunda linha:

```
pRiNtF("Linha dois");
```

Porque escrevemos o comando com letras maiúsculas.

E na linha:

```
printf("linha três")
```

Não colocamos o ";" no final da linha.

## 8.

```
/* Programa: Testes de programação
   Autor: Antonio Reis
   Função: Testes de atenção
*/
#include "stdafx.h"

int _tmain(int argc, _TCHAR* argv[])
{
    printf("Testes de programacao");

    return 0;
}
```

## 9.

```
// Capitulo002_Exercicio09.cpp : Defines the entry point for the
// console application.
//
#include "stdafx.h";
#include "stdlib.h";

#include <iostream>

using namespace std;

int _tmain(int argc, _TCHAR* argv[])
{
    cout << "Meu primeiro programa \n";

    system("PAUSE");
    return 0;
}
```

## 10.

```
// Capitulo002_Exercicio10.cpp : Defines the entry point for the
console application.
//

#include "stdafx.h"
#include "stdlib.h"

#include <iostream>

using namespace std;

int _tmain(int argc, _TCHAR* argv[])
{
    cout << "Fiz " << 2 << " programas \n";
    cout << "Letra " << 'c' << " apresentada \n";
    cout << "Frase escrita \n";
    cout << "Numero: " << 34 << " \n";
    cout << "O aumento foi de " << 10 << "% \n";

    system("PAUSE");
    return 0;
}
```

## 11.

```
// Capitulo002_Exercicio11.cpp : Defines the entry point for the
console application.
//

#include "stdafx.h"
#include "stdlib.h"

#include <iostream>

using namespace std;

int _tmain(int argc, _TCHAR* argv[])
{
    cout << "Um numero inteiro: " << 123 << "\n";
    cout << "Um numero float: " << 345.57 << "\n";
    cout << "A letra: " << 'X' << "\n";
    cout << "Um numero em formato percentual: " << 27.5 << "% \n";

    system("PAUSE");
    return 0;
}
```



**12.**

```

// Capitulo002_Exercicio12.cpp : Defines the entry point for the
console application.
//

#include "stdafx.h"
#include "stdlib.h"

#include <iostream>

using namespace std;

int _tmain(int argc, _TCHAR* argv[])
{
    cout << "Nome.....: Fulano de Tal \n";
    cout << "Endereco.....: Rua das Rosas, sn \n";
    cout << "Bairro.....: Flores \n";
    cout << "Cidade.....: Porto Alegre \n";
    cout << "Estado.....: RS \n";
    cout << "CEP.....: 90000-000 \n";

    system("PAUSE");
    return 0;
}

```

---

### Capítulo 3

---

**1.**

Regras para nomear variáveis:

Os nomes das variáveis são construídos com letras e números

- Os nomes das variáveis são construídos com letras e números
- O primeiro caractere deve ser uma letra
- Não poderá ter espaços em branco nos nomes das variáveis
- Não poderá começar com um número
- Letras maiúsculas e minúsculas são diferentes
- Os nomes devem ter um significado relacionado com o conteúdo da variável
- Nomes compostos devem começar com letras minúsculas e o início do próximo nome começa com letra maiúscula

Exemplos de nomes que são válidos:

- media
- nota1
- nota2
- dataNascimento
- salarioBruto
- taxaInss

2.

Quando declaramos uma variável ela estará apontando para uma área de memória, cujo conteúdo não é definido, portanto poderá ter qualquer valor na variável se ela não for inicializada.

3.

Como o próprio nome diz, em uma variável o conteúdo da mesma pode variar no decorrer do programa e na constante o valor se mantém fixo.

Exemplo de variável:

O valor da média de um aluno poderá variar conforme as notas no decorrer do semestre.

Podemos citar como exemplo de constante, o valor de um cálculo matemático que utiliza o valor do definido como 3.14159.

4.

Analisando o programa podemos identificar:

Nas linhas 1, 2 e 3 serão apresentados os valores das variáveis declaradas, mas que não foram inicializadas, portanto não podemos definir os valores que serão apresentados.

Nas linhas 4, 5 e 6 sabemos os valores, pois foi atribuído valores as variáveis e teremos respectivamente os valores: 10, 20 e 30;

## 5.

```
// Capitulo003_Exercicio05.cpp : Defines the entry point for the
console application.
//

#include "stdafx.h"
#include "stdlib.h"

int _tmain(int argc, _TCHAR* argv[])
{
    int kmInicial, kmFinal, distancia;

    kmInicial = 100000;
    kmFinal = 105371;

    distancia = kmFinal - kmInicial;

    printf("Distancia percorrida: %d \n", distancia);

    system("PAUSE");
    return 0;
}
```

## 6.

```
// Capitulo003_Exercicio06.cpp : Defines the entry point for the
console application.
//

#include "stdafx.h"
#include "stdlib.h"

int _tmain(int argc, _TCHAR* argv[])
{
    int nota1, nota2, nota3, nota4, media;

    nota1 = 7;
    nota2 = 3;
    nota3 = 6;
    nota4 = 5;

    media = (nota1 + nota2 + nota3 + nota4) / 4;

    printf("Media do aluno: %d \n", media);

    system("PAUSE");
    return 0;
}
```

## 7.

```
// Capitulo003_Exercicio07.cpp : Defines the entry point for the
console application.
//

#include "stdafx.h"
#include "stdlib.h"

int _tmain(int argc, _TCHAR* argv[])
{
    float nota1, nota2, nota3, nota4, media;

    nota1 = 7;
    nota2 = 3;
    nota3 = 6;
    nota4 = 5;

    media = (nota1 + nota2 + nota3 + nota4) / 4;

    printf("Media do aluno: %f \n", media);

    system("PAUSE");
    return 0;
}
```

## 8.

```
// Capitulo003_Exercicio08.cpp : Defines the entry point for the
console application.
//

#include "stdafx.h"
#include "stdlib.h"

int _tmain(int argc, _TCHAR* argv[])
{
    float volume, raio, altura;

    raio = 10;
    altura = 3;

    volume = 3.14159 * raio * raio * altura;

    printf("Volume da lata: %f \n", volume);

    system("PAUSE");
    return 0;
}
```

**9.**

```
// Capitulo003_Exercicio09.cpp : Defines the entry point for the
console application.
//

#include "stdafx.h"
#include "stdlib.h"

int _tmain(int argc, _TCHAR* argv[])
{
    float volume, comprimento, largura, altura;

    comprimento = 5;
    largura = 10;
    altura = 20;

    volume = comprimento * largura * altura;

    printf("Volume da caixa retangular: %f \n", volume);

    system("PAUSE");
    return 0;
}
```

**10.**

```
// Capitulo003_Exercicio10.cpp : Defines the entry point for the
console application.
//

#include "stdafx.h"
#include "stdlib.h"

int _tmain(int argc, _TCHAR* argv[])
{
    int valor1, valor2, valor3, valor4, valor5, distancia;
    valor1 = 120;
    valor2 = 195;
    valor3 = 47;
    valor4 = 289;
    valor5 = 38;

    distancia = valor1 + valor2 + valor3 + valor4 + valor5;

    printf("Distancia percorrida: %d \n", distancia);

    system("PAUSE");
    return 0;
}
```

## Capítulo 4

---

1.

O operador de incremento colocado do lado esquerdo faz primeiro o incremento do valor da variável para depois utilizar o valor incrementado. O operador de incremento colocado do lado direito manipula o valor da variável sem o incremento para depois fazer a operação de incremento.

2.

O operador “+=” é o operador aritmético de atribuição, que está utilizando a operação de soma.

Podemos citar o exemplo de uma variável inteira inicializada com o valor 10. Se escrevermos um comando do tipo: “variável += 20;”, ela vai utilizar o valor que já está armazenado na variável e será somado ao número 20, atribuindo o resultado a variável.

3.

Serão apresentados respectivamente os valores:

1, 2, 4, 7, 28.

4.

Serão apresentados respectivamente os valores:

0, 0, 1.

5.

Serão apresentados respectivamente os valores:

0, 1, 1.

6.

Serão apresentados respectivamente os valores:

0, 0, -1.

7.

Serão apresentados respectivamente os valores:  
10,21.

8.

Serão apresentados os valores:

10 5 7 22, 17, 7

A variável “n1” apresentada na tela tem o conteúdo: 10

A variável “n2” apresentada na tela tem o conteúdo: 5

A variável “n3” apresentada na tela tem o conteúdo: 7

A variável “n4” apresentada na tela tem o conteúdo: 22

A variável “n5” apresentada na tela tem o conteúdo: 17

A variável “n6” apresentada na tela tem o conteúdo: 7

---

## Capítulo 5

---

1.

As partes básicas de um programa são: entrada de dados, processamento de dados e a saída de dados.

2.

As perguntas que irão facilitar a nossa compreensão da lógica de um programa são:

O que temos que ler?

O que devemos fazer com os números lidos?

O que devemos mostrar?

3.

A função de biblioteca ideal para ler o nome completo de uma pessoa é a função: “get\_s”

4.

A função de biblioteca, em C++, usada para escrever uma mensagem na tela é: “cout” e a utilizada para ler um dado numérico do teclado é a função: “cin”.

## 5.

As linhas de comando que devem ser colocadas para compilar corretamente um programa em C++ são:

```
#include <iostream>
using namespace std;
```

## 6.

```
// Capitulo005_Exercicio06.cpp : Defines the entry point for the
// console application.
//

#include "stdafx.h"
#include "stdlib.h"

int _tmain(int argc, _TCHAR* argv[])
{
    char nome[80];
    float nota1, nota2, nota3, media;

    printf("Digite o nome do aluno: ");
    gets_s(nome);
    printf("Digite a primeira nota: ");
    scanf_s("%f", &nota1);
    printf("Digite a segunda nota: ");
    scanf_s("%f", &nota2);
    printf("Digite a terceira nota: ");
    scanf_s("%f", &nota3);

    media = (nota1 + nota2 + nota3) / 3;

    printf("Nome do aluno: %s \n", nome);
    printf("Media: %f \n", media);

    system("PAUSE");
    return 0;
}
```



## 7.

```
// Capitulo005_Exercicio07.cpp : Defines the entry point for the
// console application.
//

#include "stdafx.h"
#include "stdlib.h"

int _tmain(int argc, _TCHAR* argv[])
{
    float volume, raio, altura;

    printf("Digite o raio da lata: ");
    scanf_s("%f", &raio);
    printf("Digite a altura da lata: ");
    scanf_s("%f", &altura);

    volume = 3.14159 * raio * raio * altura;

    printf("Volume da lata: %f \n", volume);

    system("PAUSE");
    return 0;
}
```

## 8.

```
// Capitulo005_Exercicio08.cpp : Defines the entry point for the
// console application.
//

#include "stdafx.h"
#include "stdlib.h"

int _tmain(int argc, _TCHAR* argv[])
{
    float kmInicial, kmFinal, distancia;

    printf("Digite o valor da quilometragem inicial: ");
    scanf_s("%f", &kmInicial);
    printf("Digite o valor da quilometragem final: ");
    scanf_s("%f", &kmFinal);

    distancia = kmFinal - kmInicial;

    printf("Distancia percorrida: %f \n", distancia);

    system("PAUSE");
    return 0;
}
```

## 9.

```
// Capitulo005_Exercicio09.cpp : Defines the entry point for the
console application.
//

#include "stdafx.h"
#include "stdlib.h"

int _tmain(int argc, _TCHAR* argv[])
{
    char nome[80];
    int idade;
    float dias, horas, minutos, segundos;

    printf("Digite seu nome: ");
    gets_s(nome);
    printf("Digite a sua idade: ");
    scanf_s("%d", &idade);

    dias = idade * 365;
    horas = dias * 24;
    minutos = horas * 60;
    segundos = minutos * 60;

    printf("Voce vive, desde seu nascimento a quantidade de: \n");
    printf("%.0f dias \n", dias);
    printf("%.0f horas \n", horas);
    printf("%.0f minutos \n", minutos);
    printf("%.0f segundos \n", segundos);

    system("PAUSE");
    return 0;
}
```

**10.**

```

// Capitulo005_Exercicio10.cpp : Defines the entry point for the
console application.
//

#include "stdafx.h"
#include "stdlib.h"

int _tmain(int argc, _TCHAR* argv[])
{
    float receita, despesa, lucro;

    printf("Digite a receita da empresa: ");
    scanf_s("%f", &receita);
    printf("Digite a despesa da empresa: ");
    scanf_s("%f", &despesa);

    lucro = receita - despesa;

    printf("Lucro da empresa: %.2f \n", lucro);

    system("PAUSE");
    return 0;
}

```

---

## Capítulo 6

---

**1.**

Devemos usar um desvio condicional composto, pois se o resultado da operação for verdade diremos que é par, caso contrário diremos que é ímpar.

**2.**

O teste está sempre no if, a estrutura else não possui teste.

## 3.

Temos os seguintes operadores de comparação:

- == Igual a
- != Diferente de
- > Maior do que
- >= Maior ou igual a
- < Menor do que
- <= Menor ou igual a

## 4.

Para passar pela mensagem: “Passei por aqui” deverá ser digitado qualquer número que seja diferente de 100. Para passar pela mensagem: “Mentira passei neste lugar”, deverá ser digitado o número 100.

## 5.

```
// Capitulo006_Exercicio05.cpp : Defines the entry point for the
console application.
//

#include "stdafx.h"
#include "stdlib.h"

int _tmain(int argc, _TCHAR* argv[])
{
    int valorA, valorB;

    printf("Digite um valor: ");
    scanf_s("%d", &valorA);
    printf("Digite outro valor: ");
    scanf_s("%d", &valorB);

    if (valorA > valorB)
        printf("Valor maior: %d \n", valorA);
    else
        printf("Valor maior: %d \n", valorB);

    system("PAUSE");
    return 0;
}
```

## 6.

```
// Capitulo006_Exercicio06.cpp : Defines the entry point for the
console application.
//

#include "stdafx.h"
#include "stdlib.h"

int _tmain(int argc, _TCHAR* argv[])
{
    int valorA, valorB;

    printf("Digite um valor: ");
    scanf_s("%d", &valorA);
    printf("Digite outro valor: ");
    scanf_s("%d", &valorB);

    if (valorA < valorB)
        printf("Valor menor: %d \n", valorA);
    else
        printf("Valor menor: %d \n", valorB);

    system("PAUSE");
    return 0;
}
```

## 7.

```
// Capitulo006_Exercicio07.cpp : Defines the entry point for the
console application.
//

#include "stdafx.h"
#include "stdlib.h"

int _tmain(int argc, _TCHAR* argv[])
{
    int valorA;

    printf("Digite um valor: ");
    scanf_s("%d", &valorA);

    if (valorA == 77)
        printf("O valor e igual a 77 \n");

    system("PAUSE");
    return 0;
}
```

## 8.

```
// Capitulo006_Exercicio08.cpp : Defines the entry point for the
console application.
//

#include "stdafx.h"
#include "stdlib.h"

int _tmain(int argc, _TCHAR* argv[])
{
    int valorA;

    printf("Digite um valor: ");
    scanf_s("%d", &valorA);

    if (valorA != 123)
        printf("O valor e diferente de 123 \n");

    system("PAUSE");
    return 0;
}
```

## 9.

```
// Capitulo006_Exercicio09.cpp : Defines the entry point for the
console application.
//

#include "stdafx.h"
#include "stdlib.h"

int _tmain(int argc, _TCHAR* argv[])
{
    int valorA, valorB, quociente;

    printf("Digite um valor: ");
    scanf_s("%d", &valorA);
    printf("Digite outro valor: ");
    scanf_s("%d", &valorB);

    if (valorA > valorB)
        quociente = valorA / valorB;
    else
        quociente = valorB / valorA;
    printf("Valor maior dividido pelo valor menor: %d \n", quociente);

    system("PAUSE");
    return 0;
}
```

**10.**

```
// Capitulo006_Exercicio10.cpp : Defines the entry point for the
console application.
//

#include "stdafx.h"
#include "stdlib.h"

int _tmain(int argc, _TCHAR* argv[])
{
    float receita, despesa, lucro;

    printf("Digite a receita da empresa: ");
    scanf_s("%f", &receita);
    printf("Digite a despesa da empresa: ");
    scanf_s("%f", &despesa);

    lucro = receita - despesa;

    if (receita > despesa)
        printf("LUCRO \n");
    else
        printf("PREJUIZO \n");

    system("PAUSE");
    return 0;
}
```

**11.**

```
// Capitulo006_Exercicio11.cpp : Defines the entry point for the
console application.
//

#include "stdafx.h"
#include "stdlib.h"

int _tmain(int argc, _TCHAR* argv[])
{
    float nota1, nota2, nota3, nota4, media;

    printf("Digite a primeira nota: ");
    scanf_s("%f", &nota1);
    printf("Digite a segunda nota: ");
    scanf_s("%f", &nota2);
    printf("Digite a terceira nota: ");
    scanf_s("%f", &nota3);
    printf("Digite a quarta nota: ");
    scanf_s("%f", &nota4);
```

```

media = (nota1 + nota2 + nota3 + nota4) / 4;

printf("Media: %f \n", media);
if (media >= 6)
    printf("APROVADO \n");
else
    printf("REPROVADO \n");

system("PAUSE");
return 0;
}

```

**12.**

```

// Capitulo006_Exercicio12.cpp : Defines the entry point for the
console application.
//

#include "stdafx.h"
#include "stdlib.h"

int _tmain(int argc, _TCHAR* argv[])
{
    int valorA, valorB, produto;

    printf("Digite um valor: ");
    scanf_s("%d", &valorA);
    printf("Digite outro valor: ");
    scanf_s("%d", &valorB);

    produto = valorA * valorB;

    if (produto >= 0)
        printf("Sinal do produto: POSITIVO \n");
    else
        printf("Sinal do produto: NEGATIVO \n");

    system("PAUSE");
    return 0;
}

```



**13.**

```
// Capitulo006_Exercicio13.cpp : Defines the entry point for the
console application.
//

#include "stdafx.h"
#include "stdlib.h"

int _tmain(int argc, _TCHAR* argv[])
{
    int numero;

    printf("Digite um numero: ");
    scanf_s("%d", &numero);

    if (numero > 1000)
        printf("Maior do que 1000 \n");
    else
        printf("Menor ou igual a 1000 \n");

    system("PAUSE");
    return 0;
}
```

**14.**

```
// Capitulo0064_Exercicio01.cpp : Defines the entry point for the
console application.
//

#include "stdafx.h"
#include "stdlib.h"

int _tmain(int argc, _TCHAR* argv[])
{
    int horasTrab, filhos;
    float salHora, salBruto, salFamilia, valor;

    printf("Digite o valor do salario hora: ");
    scanf_s("%f", &salHora);
    printf("Digite a quantidade de horas trabalhadas: ");
    scanf_s("%d", &horasTrab);
    printf("Digite a quantidade de filhos menores de 14 anos: ");
    scanf_s("%d", &filhos);

    salBruto = horasTrab * salHora;

    if (salBruto <= 700)
        valor = 9.5;
```

```

else if (salBruto <= 1000)
    valor = 6.6;
else
    valor = 1.5;

salFamilia = filhos * valor;

printf("Salario Bruto....: %.2f \n", salBruto);
printf("Salario Familia..: %.2f \n", salFamilia);

system("PAUSE");
return 0;
}

```

## 15.

```

// Capitulo0064_Exercicio02.cpp : Defines the entry point for the
// console application.
//
#include "stdafx.h"
#include "stdlib.h"

int _tmain(int argc, _TCHAR* argv[])
{
    float salBruto, desconto, taxa, salLiquido;

    printf("Digite o valor do salario bruto: ");
    scanf_s("%f", &salBruto);

    if (salBruto <= 1317.07)
        taxa = 8;
    else if (salBruto <= 2195.12)
        taxa = 9;
    else
        taxa = 11;

    desconto = salBruto * (taxa / 100);

    if (desconto > 482.93)
        desconto = 482.93;

    salLiquido = salBruto - desconto;

    printf("Salario Bruto....: %.2f \n", salBruto);
    printf("Taxa de INSS....: %.2f \n", taxa);
    printf("Desconto de INSS.: %.2f \n", desconto);
    printf("Salario Liquido...: %.2f \n", salLiquido);

    system("PAUSE");
    return 0;
}

```

**16.**

```
// Capitulo0064_Exercicio03.cpp : Defines the entry point for the
console application.
//

#include "stdafx.h"
#include "stdlib.h"

int _tmain(int argc, _TCHAR* argv[])
{
    int codigo;

    printf("Digite o codigo numerico do produto: ");
    scanf_s("%d", &codigo);

    if (codigo == 1)
        printf("Chaleira \n");
    else if (codigo == 2)
        printf("Prato \n");
    else if (codigo == 3)
        printf("Panela \n");

    system("PAUSE");
    return 0;
}
```

**17.**

A estrutura de desvio condicional para resolver o problema é chamada de Desvio Condicional de Múltiplos Casos. Como as comparações são para saber se é igual a 10, se é igual a 11, se é igual a 12 e assim sucessivamente até o valor 20.

**18.**

A estrutura de desvio condicional para resolver este problema é a chamada de Desvio Condicional Encadeado, pois fazemos o teste escalonado para cada faixa de salário. Primeiro faz o teste para a primeira faixa, depois o da segunda faixa e assim sucessivamente.

**19.**

```
Código fonte do programa:
// Capitulo0066_Exercicio03.cpp : Defines the entry point for the
console application.
//
#include "stdafx.h"
```

```
#include "stdlib.h"
#include "conio.h"

int _tmain(int argc, _TCHAR* argv[])
{
    char letra;

    printf("Digite uma letra em maiusculo: ");
    letra = _getche();

    printf("\n\n");

    switch (letra)
    {
    case 'A':
        printf("A letra digitada foi: A \n");
        break;
    case 'B':
        printf("A letra digitada foi: B \n");
        break;
    case 'C':
        printf("A letra digitada foi: C \n");
        break;
    case 'D':
        printf("A letra digitada foi: D \n");
        break;
    case 'E':
        printf("A letra digitada foi: E \n");
        break;
    case 'F':
        printf("A letra digitada foi: F \n");
        break;
    case 'G':
        printf("A letra digitada foi: G \n");
        break;
    case 'H':
        printf("A letra digitada foi: H \n");
        break;
    case 'I':
        printf("A letra digitada foi: I \n");
        break;
    case 'J':
        printf("A letra digitada foi: J \n");
        break;
    case 'K':
        printf("A letra digitada foi: K \n");
        break;
    case 'L':
        printf("A letra digitada foi: L \n");
        break;
    case 'M':
        printf("A letra digitada foi: M \n");
```

```
        break;
    case 'N':
        printf("A letra digitada foi: N \n");
        break;
    case 'O':
        printf("A letra digitada foi: O \n");
        break;
    case 'P':
        printf("A letra digitada foi: P \n");
        break;
    case 'Q':
        printf("A letra digitada foi: Q \n");
        break;
    case 'R':
        printf("A letra digitada foi: R \n");
        break;
    case 'S':
        printf("A letra digitada foi: S \n");
        break;
    case 'T':
        printf("A letra digitada foi: T \n");
        break;
    case 'U':
        printf("A letra digitada foi: U \n");
        break;
    case 'V':
        printf("A letra digitada foi: V \n");
        break;
    case 'X':
        printf("A letra digitada foi: X \n");
        break;
    case 'Y':
        printf("A letra digitada foi: Y \n");
        break;
    case 'Z':
        printf("A letra digitada foi: Z \n");
        break;
    default:
        printf("Voce deve digitar uma letra maiuscula \n");
        break;
}

system("\nPAUSE");
return 0;
}
```

---

## Capítulo 7

---

1.

Devemos usar o conector lógico “&&”, pois o teste que deve ser realizado é para saber se a nota é  $\geq 0$  e se a nota é  $\leq 10$ . Para saber se a nota está dentro da faixa de números temos que usar o conector lógico “&&”.

2.

Neste caso vamos usar o conector lógico “||”. Iremos verificar se o código digitado foi igual a 1 ou se o código digitado foi igual a 2 ou se foi igual a 3, caso contrário o código digitado é inválido.

3.

Tabela com as Mensagens apresentadas conforme as escolhas do usuário:

Variáveis			Mensagem que Será apresentada			
Café	Almoço	Jantar	Todas as refeições devem ser preparadas	Uma ou mais refeições deve ser preparada	Nenhuma refeição deve ser preparada	Código digitado invalido, o programa será encerrado
0	0	0	( )	( )	(x)	( )
0	1	0	( )	(x)	( )	( )
0	0	1	( )	(x)	( )	( )
1	0	0	( )	(x)	( )	( )
1	1	1	(x)	( )	( )	( )
1	0	1	( )	(x)	( )	( )
1	1	0	( )	(x)	( )	( )
0	1	1	( )	(x)	( )	( )
1	0	1	( )	(x)	( )	( )
10	0	0	( )	( )	( )	(x)
0	10	0	( )	( )	( )	(x)
0	0	10	( )	( )	( )	(x)
10	10	10	( )	( )	( )	(x)
99	88	98	( )	( )	( )	(x)

## 4.

```

// Capitulo007_Exercicio04.cpp : Defines the entry point for the
console application.
//

#include "stdafx.h"
#include "stdlib.h"

int _tmain(int argc, _TCHAR* argv[])
{
    float ladoA, ladoB, ladoC;

    printf("Digite o valor de um dos lados do triangulo: ");
    scanf_s("%f", &ladoA);
    printf("Digite o valor de outro lado do triangulo: ");
    scanf_s("%f", &ladoB);
    printf("Digite o valor do ultimo lado do triangulo: ");
    scanf_s("%f", &ladoC);

    if (ladoA < (ladoB + ladoC) && ladoB < (ladoA + ladoC) && ladoC <
(ladoA + ladoB))
    {
        if (ladoA == ladoB && ladoA == ladoC)
            printf("Triangulo equilatero \n");
        else if (ladoA != ladoB && ladoB != ladoC)
            printf("Triangulo escaleno \n");
        else
            printf("Triangulo isosceles \n");
    }
    else
    {
        printf("Os lados não formam um triangulo \n");
    }

    system("PAUSE");
    return 0;
}

```

## 5.

```
// Capitulo007_Exercicio05.cpp : Defines the entry point for the
console application.
//

#include "stdafx.h"
#include "stdlib.h"

int _tmain(int argc, _TCHAR* argv[])
{
    int x, y;

    printf("Digite o valor da coordenada x: ");
    scanf_s("%d", &x);
    printf("Digite o valor da coordernada y: ");
    scanf_s("%d", &y);

    if (x > 0 && y > 0)
        printf("Primeiro quadrante \n");
    else if (x < 0 && y > 0)
        printf("Segundo quadrante \n");
    else if (x < 0 && y < 0)
        printf("Terceiro quadrante \n");
    else if (x > 0 && y < 0)
        printf("Quarto quadrante \n");
    else
        printf("Uma das coordenadas esta na origem \n");

    system("PAUSE");
    return 0;
}
```



## 6.

```
// Capitulo007_Exercicio06.cpp : Defines the entry point for the
console application.
//

#include "stdafx.h"
#include "stdlib.h"

int _tmain(int argc, _TCHAR* argv[])
{
    float qi;

    printf("Digite o coeficiente de inteligencia: ");
    scanf_s("%f", &qi);

    if (qi >= 0 && qi < 30)
        printf("Acefalo \n");
    else if (qi >= 30 && qi < 50)
        printf("Debil mental \n");
    else if (qi >= 50 && qi < 70)
        printf("Regular \n");
    else if (qi >= 70 && qi < 100)
        printf("Normal \n");
    else if (qi >= 100 && qi < 150)
        printf("Genio \n");
    else
        printf("Valor digitado errado \n");

    system("PAUSE");
    return 0;
}
```

## 7.

```
// Capitulo007_Exercicio07.cpp : Defines the entry point for the
console application.
//

#include "stdafx.h"
#include "stdlib.h"

int _tmain(int argc, _TCHAR* argv[])
{
    char nome[80];
    float nota1, nota2, nota3, media;

    printf("Digite o nome do aluno: ");
    gets_s(nome);
    printf("Digite a primeira nota: ");
    scanf_s("%f", &nota1);
    printf("Digite a segunda nota: ");
    scanf_s("%f", &nota2);
    printf("Digite a terceira nota: ");
    scanf_s("%f", &nota3);

    media = (nota1 + nota2 + nota3) / 3;

    printf("Nome do aluno: %s \n", nome);
    printf("Media: %f \n", media);

    if (media >= 0 && media < 4)
        printf("Reprovado \n");
    else if (media >= 4 && media < 6)
        printf("Recuperacao \n");
    else if (media >= 6 && media < 9)
        printf("Satisfatorio \n");
    else if (media >= 9 && media < 10)
        printf("Plenamente satisfatorio \n");
    else
        printf("Voce digitou as notas fora da faixa de 0 ate 10 \n");

    system("PAUSE");
    return 0;
}
```

---

## Capítulo 8

---

### 1.

Na linha 1 devemos colocar a inicialização da variável contadora:

```
contar = 0;
```

Na linha 2 para controlar o laço de repetição colocamos:

```
while (contar < 20)
```

Na linha 3, vamos reinicializar a variável contadora:

```
contar = contar + 1;
```

### 2.

Código fonte do programa:

```
// Capitulo0081_Exercicio02.cpp : Defines the entry point for the
// console application.
//

#include "stdafx.h"
#include "stdlib.h"

int _tmain(int argc, _TCHAR* argv[])
{
    int cont;

    cont = 0;
    while (cont <= 20)
    {
        printf("%d \n", cont);
        cont = cont + 1;
    }

    system("PAUSE");
    return 0;
}
```

## 3.

```
// Capitulo0081_Exercicio03.cpp : Defines the entry point for the
console application.
//

#include "stdafx.h"
#include "stdlib.h"

int _tmain(int argc, _TCHAR* argv[])
{
    int cont, numero, soma;

    soma = cont = 0;
    while (cont < 10)
    {
        printf("Digite um numero: ");
        scanf_s("%d", &numero);

        soma = soma + numero;

        cont = cont + 1;
    }
    printf("Somatorio dos numeros lidos: %d \n", soma);

    system("PAUSE");
    return 0;
}
```

## 4.

```
// Capitulo0081_Exercicio04.cpp : Defines the entry point for the
console application.
//

#include "stdafx.h"
#include "stdlib.h"

int _tmain(int argc, _TCHAR* argv[])
{
    int cont, numero, soma, media, contNeg;

    soma = media = contNeg = cont = 0;
    while (cont < 20)
    {
        printf("Digite um numero: ");
        scanf_s("%d", &numero);

        if (numero >= 0)
            soma = soma + numero;
        else
```

```

    {
        media = media + numero;
        contNeg = contNeg + 1;
    }
    cont = cont + 1;
}
printf("Somatorio dos numeros positivos: %d \n", soma);
media = media / contNeg;
printf("Media dos numeros negativos: %d \n", media);

system("PAUSE");
return 0;
}

```

## 5.

```

// Capitulo0082_Exercicio01.cpp : Defines the entry point for the
// console application.
//

#include "stdafx.h"
#include "stdlib.h"

int _tmain(int argc, _TCHAR* argv[])
{
    int numero, soma, media, contNum;
    char resposta = 's';

    soma = contNum = 0;

    while (resposta == 's')
    {
        printf("Digite um numero: ");
        scanf_s("%d", &numero);
        soma += numero;
        contNum++;

        printf("Deseja digitar outro numero? (s/n) ");
        resposta = _gettche();
        printf("\n\n");
    }

    media = soma / contNum;
    printf("Media dos valores lidos: %d \n", media);

    system("PAUSE");
    return 0;
}

```

## 6.

```
// Capitulo0082_Exercicio02.cpp : Defines the entry point for the
console application.
//

#include "stdafx.h"
#include "stdlib.h"

int _tmain(int argc, _TCHAR* argv[])
{
    int numero, somaPos, media, contNum, qtdZeros;
    char resposta = 's';

    somaPos = media = contNum = qtdZeros = 0;

    while (resposta == 's')
    {
        printf("Digite um numero: ");
        scanf_s("%d", &numero);

        if (numero > 0)
            somaPos += numero;
        else if (numero == 0)
            qtdZeros++;

        media = media + numero;
        contNum++;

        printf("Deseja digitar outro numero? (s/n) ");
        resposta = _gettche();
        printf("\n\n");
    }

    printf("Somatorio dos Positivos: %d \n", somaPos);
    printf("Quantidade de zeros....: %d \n", qtdZeros);
    media = media / contNum;
    printf("Media dos valores lidos: %d \n", media);

    system("PAUSE");
    return 0;
}
```

## 7.

```

// Capitulo0082_Exercicio03.cpp : Defines the entry point for the
console application.
//

#include "stdafx.h"
#include "stdlib.h"

int _tmain(int argc, _TCHAR* argv[])
{
    int idade, codigo;
    int mediaSolteiros, mediaCasados, mediaDivorciados;
    int contaSolteiros, contaCasados, contaDivorciados;
    char resposta = 's';

    mediaSolteiros = mediaCasados = mediaDivorciados = 0;
    contaSolteiros = contaCasados = contaDivorciados = 0;

    while (resposta == 's')
    {
        system("CLS");
        printf("Digite a idade da pessoa: ");
        scanf_s("%d", &idade);

        printf("Escolha o codigo do estado civil \n");
        printf("1 - Solteiro \n");
        printf("2 - Casado \n");
        printf("3 - Divorciado \n");
        printf("Escolha um codigo: ");
        scanf_s("%d", &codigo);

        switch (codigo)
        {
        case 1:
            mediaSolteiros += idade;
            contaSolteiros++;
            break;
        case 2:
            mediaCasados += idade;
            contaCasados++;
            break;
        case 3:
            mediaDivorciados += idade;
            contaDivorciados++;
            break;
        default:
            printf("Codigo digitado não existe");
            break;
        }
    }
}

```

```

printf("Deseja digitar outro numero? (s/n) ");
resposta = _gettche();
printf("\n\n");
}

mediaSolteiros /= contaSolteiros;
mediaCasados /= contaCasados;
mediaDivorciados /= contaDivorciados;

printf("Media da idade dos solteiros...: %d \n", mediaSolteiros);
printf("Media da idade dos casados....: %d \n", mediaCasados);
printf("Media da idade dos divorciados: %d \n", mediaDivorciados);

system("PAUSE");
return 0;
}

```

**8.**

O laço de repetição for possui 3 partes:

A primeira é a inicialização da variável contadora

A segunda é teste de repetição

A terceira é a reinicialização da variável contadora.

**9.**

Poderíamos usar qualquer uma das estruturas citadas: while ou for, mas a estrutura mais indicada para um laço de repetição com um quantidade determinada de repetições (1200 funcionários) é o for.

**10.**

Iríamos colocar a seguinte estrutura:

```
for( contar = 0; contar < 120; contar++)
```

**11.**

O resultado na tela do programa será a apresentação de números, começando em 0, 2, 4, 6, 8, ... até 100. Será apresentado um em cada linha. Poderíamos traduzir como o programa que escreve na tela os números pares de 0 até 100.



**12.**

O resultado na tela do programa será a apresentação de números, começando em 100, 98, 96, 94, ... até 0. Será apresentado um em cada linha. Poderíamos traduzir como o programa que escrever na tela os números pares de 100 até 0.

**13.**

```
// Capitulo0083_Exercicio06.cpp : Defines the entry point for the
console application.
//

#include "stdafx.h"
// Capitulo0082_Exercicio03.cpp : Defines the entry point for the
console application.
//

#include "stdafx.h"
#include "stdlib.h"

int _tmain(int argc, _TCHAR* argv[])
{
    int cont;

    for (cont = 0; cont <= 20; cont++)
    {
        printf("%d \n", cont);
    }

    system("PAUSE");
    return 0;
}
```

**14.**

```
// Capitulo0083_Exercicio07.cpp : Defines the entry point for the
console application.
//
#include "stdafx.h"
#include "stdlib.h"

int _tmain(int argc, _TCHAR* argv[])
{
    int cont, soma = 0;

    for (cont = 0; cont <= 100; cont++)
    {
        soma += cont;
    }
}
```

```

printf("O somatorio dos numeros: %d \n", soma);

system("PAUSE");
return 0;
}

```

## 15.

```

// Capitulo0083_Exercicio08.cpp : Defines the entry point for the
console application.
//

#include "stdafx.h"
#include "stdlib.h"

int _tmain(int argc, _TCHAR* argv[])
{
    int cont, somaPos, somaNeg, contNeg, mediaNeg, numero;

    somaPos = somaNeg = contNeg = 0;

    for (cont = 0; cont < 20; cont++)
    {
        printf("Digite um numero: ");
        scanf_s("%d", &numero);

        if (numero >= 0)
            somaPos += numero;
        else
        {
            somaNeg += numero;
            contNeg++;
        }
    }

    if (contNeg != 0)
    {
        mediaNeg = somaNeg / contNeg;
        printf("Media dos numeros negativos: %d \n", mediaNeg);
    }
    else
        printf("Voce nao digitou numeros negativos \n");

    printf("O somatorio dos numeros positivos: %d \n", somaPos);

    system("PAUSE");
    return 0;
}

```

## 16.

```
// Capitulo0083_Exercicio09.cpp : Defines the entry point for the
console application.
//

#include "stdafx.h"
// Capitulo0083_Exercicio08.cpp : Defines the entry point for the
console application.
//

#include "stdafx.h"
#include "stdlib.h"

int _tmain(int argc, _TCHAR* argv[])
{
    int cont, num1, num2;

    printf("Digite um numero: ");
    scanf_s("%d", &num1);
    printf("Digite outro numero: ");
    scanf_s("%d", &num2);

    if (num1 < num2)
        for (cont = num1; cont <= num2; cont++)
            printf("%d \n", cont);
    else
        for (cont = num1; cont >= num2; cont--)
            printf("%d \n", cont);

    system("PAUSE");
    return 0;
}
```

## 17.

```
// Capitulo0084_Exercicio01.cpp : Defines the entry point for the
console application.
//

#include "stdafx.h"
#include "stdlib.h"

int _tmain(int argc, _TCHAR* argv[])
{
    int codigo, estoque, totalEntrada, totalSaida;

    estoque = totalEntrada = totalSaida = 0;
```

```

do {
    system("CLS");
    printf("Menu de opcoes \n");
    printf("0 - Encerrar o programa \n");
    printf("1 - Incluir uma unidade no estoque \n");
    printf("2 - Excluir uma unidade no estoque \n");
    printf("3 - Mostrar o total disponivel no estoque \n");
    printf("4 - Mostrar o total de entradas no estoque \n");
    printf("5 - Mostrar o total de saidas no estoque \n");
    printf("Escolha uma opcao: ");
    scanf_s("%d", &codigo);

    switch (codigo)
    {
    case 0:
        printf("Encerrando o programa \n");
        break;
    case 1:
        estoque++;
        totalEntrada++;
        printf("Incluindo uma unidade no estoque \n");
        break;
    case 2:
        if (estoque == 0)
            printf("Estoque esta vazio \n");
        else
        {
            printf("Excluindo uma unidade no estoque \n");
            estoque--;
            totalSaida++;
        }
        break;
    case 3:
        printf("Total disponivel no estoque: %d \n", estoque);
        break;
    case 4:
        printf("Total de entradas no estoque: %d \n", totalEntrada);
        break;
    case 5:
        printf("Total de saidas no estoque: %d \n", totalSaida);
        break;
    default:
        printf("Voce digitou o codigo invalido \n");
        break;
    }
    system("PAUSE");
} while (codigo != 0);

return 0;
}

```

## 18.

```

// Capitulo0084_Exercicio02.cpp : Defines the entry point for the
// console application.
//

#include "stdafx.h"
#include "stdlib.h"

int _tmain(int argc, _TCHAR* argv[])
{
    int codigo, disponivel, totalEntrada, totalSaida, capacidade;

    disponivel = capacidade = 5;
    totalEntrada = totalSaida = 0;

    do {
        system("CLS");
        printf("Menu de opcoes \n");
        printf("0 - Encerrar o programa \n");
        printf("1 - Entrar uma pessoa \n");
        printf("2 - Sair uma pessoa \n");
        printf("3 - Mostrar quantos lugares estao disponiveis \n");
        printf("4 - Mostrar o total de pessoas que entraram \n");
        printf("5 - Mostrar o total de pessoas que sairam \n");
        printf("Escolha uma opcao: ");
        scanf_s("%d", &codigo);

        switch (codigo)
        {
            case 0:
                printf("Encerrando o programa \n");
                break;
            case 1:
                if (disponivel > 0)
                {
                    disponivel--;
                    totalEntrada++;
                    printf("Entrou uma pessoa \n");
                }
                else
                    printf("Lancheria esta lotada, aguarde \n");
                break;
            case 2:
                if (disponivel < capacidade)
                {
                    disponivel++;
                    totalSaida++;
                    printf("Saiu uma pessoa \n");
                }
        }
    }
}

```

```

        else
            printf("Lancheria esta vazia \n");
            break;
        case 3:
            printf("Total de lugares disponivel: %d \n", disponivel);
            break;
        case 4:
            printf("Total de pessoas que entraram: %d \n", totalEntrada);
            break;
        case 5:
            printf("Total de pessoas que sairam: %d \n", totalSaida);
            break;
        default:
            printf("Voce digitou o codigo invalido \n");
            break;
    }
    system("PAUSE");
} while (codigo != 0);

return 0;
}

```

## 19.

```

// Capitulo0084_Exercicio03.cpp : Defines the entry point for the
// console application.
//

#include "stdafx.h"
#include "stdlib.h"

int _tmain(int argc, _TCHAR* argv[])
{
    int quantidade, peso, totalPeso, lotado;

    lotado = quantidade = totalPeso = 0;

    do {
        system("CLS");
        printf("Digite o peso da pessoa que quer entrar no elevador: ");
        scanf_s("%d", &peso);

        totalPeso += peso;
        quantidade++;

        if (quantidade >= 12 || totalPeso >= 700)
        {
            printf("LOTADO \n");
            lotado = 1;
            if (quantidade > 12)

```

```

    {
        quantidade--;
        totalPeso -= peso;
    }
    if (peso > 700)
    {
        quantidade--;
        totalPeso -= peso;
    }
}
else
printf("Pode entrar \n");

system("PAUSE");

} while (lotado == 0);

printf("Total de pessoas no elevador: %d \n", quantidade);
printf("Total de peso no elevador...: %d \n", totalPeso);

system("PAUSE");
return 0;
}

```

---

## Capítulo 8

---

1.

Sim, podemos armazenar em um vetor, pois os conteúdos a serem armazenados são do mesmo tipo de informação e serão do mesmo tipo de dado.

2.

A declaração do vetor será da seguinte forma:  
int vetor[20000];

3.

As linhas de comandos para controle do laço de repetição ler um vetor de 20000 número inteiros está apresentado nas linhas abaixo:

```

for(int i = 0; i < 20000; i++)
{
    printf("Digite um numero: ");
    scanf_s("%d", &vetor[i]);
}

```

## 4.

```

// Capitulo 9
//

#include "stdafx.h"
#include "stdlib.h"

int _tmain(int argc, _TCHAR* argv[])
{
float numeros[10];

for (int i = 0; i < 10; i++)
{
printf("Digite um numero: ");
scanf_s("%f", &numeros[i]);
}

for (int i = 0; i < 10; i++)
{
printf("Numero digitado: %f \n", numeros[i]);
}

system("PAUSE");
return 0;
}

```

## 5.

```

// Capitulo0091_Exercicio05.cpp : Defines the entry point for the
console application.
//

#include "stdafx.h"
#include "stdlib.h"

int _tmain(int argc, _TCHAR* argv[])
{
int vetorA[20], vetorB[20], vetorC[20];

for (int i = 0; i < 20; i++)
{
printf("Digite o elemento %d do vetorA: ", i + 1);
scanf_s("%d", &vetorA[i]);
printf("Digite o elemento %d do vetorB: ", i + 1);
scanf_s("%d", &vetorB[i]);

vetorC[i] = vetorA[i] + vetorB[i];
}
}

```



```

for (int i = 0; i < 20; i++)
    printf("%d + %d = %d \n",vetorA[i], vetorB[i], vetorC[i]);

system("PAUSE");
return 0;
}

```

## 6.

```

// Capitulo0091_Exercicio06.cpp : Defines the entry point for the
console application.
//

#include "stdafx.h"
#include "stdlib.h"

int _tmain(int argc, _TCHAR* argv[])
{
    int vetorA[20], vetorB[20];

    for (int i = 0; i < 20; i++)
    {
        printf("Digite o elemento %d do vetorA: ", i + 1);
        scanf_s("%d", &vetorA[i]);

        vetorB[i] = vetorA[i] * vetorA[i];
    }

    for (int i = 0; i < 20; i++)
        printf("%d elevado ao quadrado %d \n", vetorA[i], vetorB[i]);

    system("PAUSE");
    return 0;
}

```

## 7.

```

// Capitulo91_Exercicio07.cpp : Defines the entry point for the console
application.
//

#include "stdafx.h"
#include "stdlib.h"

int _tmain(int argc, _TCHAR* argv[])
{
    int vetorA[20], vetorB[20], vetorC[40];

    for (int i = 0; i < 20; i++)

```

```

{
    printf("Digite o elemento %d do vetorA: ", i + 1);
    scanf_s("%d", &vetorA[i]);
    printf("Digite o elemento %d do vetorB: ", i + 1);
    scanf_s("%d", &vetorB[i]);

    vetorC[i] = vetorA[i];
    vetorC[i + 20] = vetorB[i];
}

for (int i = 0; i < 40; i++)
    printf("Elemento %d do vetorC %d \n", i + 1, vetorC[i]);

system("PAUSE");
return 0;
}

```

**8.**

```

// Capitulo0091_Exercicio08.cpp : Defines the entry point for the
// console application.
//
#include "stdafx.h"
#include "stdlib.h"

int _tmain(int argc, _TCHAR* argv[])
{
    int vetorA[20], vetorB[20];

    for (int i = 0; i < 20; i++)
    {
        printf("Digite o elemento %d do vetorA: ", i);
        scanf_s("%d", &vetorA[i]);
    }

    for (int i = 0; i < 20; i++)
    {
        vetorB[i] = vetorA[19 - i];
    }

    for (int i = 0; i < 20; i++)
        printf("Elemento %d do vetorB %d \n", i, vetorB[i]);

    system("PAUSE");
    return 0;
}

```

**9.**

```
// Capitulo0091_Exercicio09.cpp : Defines the entry point for the
console application.
//

#include "stdafx.h"
#include "stdlib.h"

int _tmain(int argc, _TCHAR* argv[])
{
    int vetorA[10];

    for (int i = 0; i < 10; i++)
    {
        vetorA[i] = 3 * i;
    }

    printf("Tabuada do numero 3 \n");
    for (int i = 0; i < 10; i++)
        printf("3 * %d = %d \n", i, vetorA[i]);

    system("PAUSE");
    return 0;
}
```

**10.**

```
// Capitulo0091_Exercicio10.cpp : Defines the entry point for the
console application.
//

#include "stdafx.h"
#include "stdlib.h"

int _tmain(int argc, _TCHAR* argv[])
{
    float minimas[30], maximas[30];
    float maior, menor;
    int diaMaior, diaMenor;

    for (int i = 0; i < 30; i++)
    {
        printf("Digite a menor temperatura do dia %d: ", i + 1);
        scanf_s("%f", &minimas[i]);
        printf("Digite a maior temperatura do dia %d: ", i + 1);
        scanf_s("%f", &maximas[i]);
    }
}
```

```

menor = minimas[0]; //inicializando variáveis
maior = maximas[0];
diaMenor = 0;
diaMaior = 0;

for (int i = 0; i < 30; i++)
{
    if (minimas[i] < menor)
    {
        menor = minimas[i];
        diaMenor = i;
    }
    if (maximas[i] > maior)
    {
        maior = maximas[i];
        diaMaior = i;
    }
}

printf("A menor temperatura foi: %f \n", menor);
printf("E ocorreu no dia: %d \n", diaMenor + 1);
printf("A maior temperatura foi: %f \n", maior);
printf("E ocorreu no dia: %d \n", diaMaior + 1);

system("PAUSE");
return 0;
}

```

**11.**

```

// Capitulo0092_Exercicio01.cpp : Defines the entry point for the
// console application.
//
#include "stdafx.h"
#include "stdlib.h"

int _tmain(int argc, _TCHAR* argv[])
{
    int matrizA[5][3], matrizB[5][3], matrizC[5][3];
    for (int i = 0; i < 5; i++)
        for (int j = 0; j < 3; j++)
        {
            printf("Digite o elemento da ");
            printf("linha %d coluna %d da matriz A: ", i + 1, j + 1);
            scanf_s("%d", &matrizA[i][j]);

            printf("Digite o elemento da ");
            printf("linha %d coluna %d da matriz B: ", i + 1, j + 1);
            scanf_s("%d", &matrizB[i][j]);
            matrizC[i][j] = matrizA[i][j] + matrizB[i][j];
        }
}

```

```

    for (int i = 0; i < 5; i++)
        for (int j = 0; j < 3; j++)
        {
            printf("Elemento da linha ");
            printf("%d coluna %d da matriz: %d \n", i + 1, j + 1, matrizC[i
[j]]);
        }

    system("PAUSE");
    return 0;
}

```

## 12.

```

// Capitulo0092_Exercicio02.cpp : Defines the entry point for the
console application.
//

#include "stdafx.h"
#include "stdlib.h"

int _tmain(int argc, _TCHAR* argv[])
{
    int matrizA[6][6], soma = 0;

    for (int i = 0; i < 6; i++)
        for (int j = 0; j < 6; j++)
        {
            printf("Digite o elemento da ");
            printf("linha %d coluna %d da matriz A : ", i, j);
            scanf_s("%d", &matrizA[i][j]);

            if (i == j)
                soma += matrizA[i][j];
        }

    printf("Soma da diagonal principal: %d \n", soma);

    system("PAUSE");
    return 0;
}

```

---

## Capítulo 10

---

1.

Um dos motivos para a utilização de funções é a manutenção do programa que fica mais fácil. Outro motivo é para diminuir o código fonte do programa, pois quando percebermos que temos muitas linhas de código que se repetem com frequência, transformamos o código em uma função e aí fazemos a chamada da função no lugar onde os códigos são repetidos.

2.

Um procedimento e uma função são um bloco de comandos com início e fim definido, identificado por um nome. Através deste nome, este bloco de comandos poderá ser chamado em qualquer parte do programa. A única diferença é que uma função pode retornar um valor para a parte do programa que a chamou e o procedimento não retorna nenhum valor.

3.

Na programação estruturada as funções são chamadas de funções, mas na técnica da programação orientada a objeto elas são chamadas de métodos.

4.

O protótipo da função: “calcular” identificada na linha 1 é:

```
void calcular(void);
```

Na linha 2 que é a declaração da função será escrito:

```
void calcular(void)
```

onde:

A palavra “void” identifica que não teremos valor de retorno e que não teremos passagem de parâmetros. Ou seja, void significa que não tem nenhum valor no local onde foi colocada.

## 5.

```

// Capitulo0104_Exercicio01.cpp : Defines the entry point for the
console application.
//

#include "stdafx.h"
#include "stdlib.h"

//Área de declaração dos protótipos das funções
void circulo();
void retangulo();
void quadrado();

int _tmain(int argc, _TCHAR* argv[])
{
    int opcao;

    do
    {
        //chave inicial do laço de repetição
        system("CLS");
        printf("Menu de opcoes\n");
        printf("1 - Area do circulo \n");
        printf("2 - Area do retangulo \n");
        printf("3 - Area do quadrado \n");
        printf("4 - Encerrar o programa \n");
        printf("Escolha uma opcao: ");
        scanf_s("%d", &opcao);

        system("CLS");
        switch (opcao)
        {
            //chave inicial do desvio condicional
            case 1:
                circulo();           //chamada da função circulo
                break;
            case 2:
                retangulo();         //chamada da função retangulo
                break;
            case 3:
                quadrado();          //chamada da função quadrado
                break;
            case 4:
                printf("Encerrando o programa \n");
                break;
            default:
                printf("Opcao invalida \n");
                break;
        }
        //chave final do desvio condicional

        system("PAUSE");
    }
}

```





## 6.

```

// Capitulo0104_Exercicio06.cpp : Defines the entry point for the
console application.
//
#include "stdafx.h"
#include "stdlib.h"

//Área de declaração dos protótipos das funções
void lata();
void caixa();
void piscina();

int _tmain(int argc, _TCHAR* argv[])
{
    int opcao;

    do
    {
        //chave inicial do laço de repetição
        system("CLS");
        printf("Menu de opcoes\n");
        printf("1 - Volume lata cilindrica \n");
        printf("2 - Volume caixa quadrada \n");
        printf("3 - Volume piscina \n");
        printf("4 - Encerrar o programa \n");
        printf("Escolha uma opcao: ");
        scanf_s("%d", &opcao);

        system("CLS");
        switch (opcao)
        {
            //chave inicial do desvio condicional
            case 1:
                lata();
                break;
            case 2:
                caixa();
                break;
            case 3:
                piscina();
                break;
            case 4:
                printf("Encerrando o programa \n");
                break;
            default:
                printf("Opcao invalida \n");
                break;
        }
        //chave final do desvio condicional

        system("PAUSE");

    } while (opcao != 4);
        //chave final do laço de repetição

```

```

    return 0;
} //chave final da função main

//Área de declaração de funções

//Declaração da função lata
void lata()
{
    float raio, altura, volume;

    printf("Volume da lata \n\n");
    printf("Digite o valor do raio: ");
    scanf_s("%f", &raio);
    printf("Digite o valor da altura: ");
    scanf_s("%f", &altura);

    volume = 3.14159 * raio * raio * altura;

    printf("\nVolume da lata: %f \n\n", volume);
}

//Declaração da função caixa
void caixa()
{
    float lado, volume;

    printf("Volume da caixa quadrada \n\n");
    printf("Digite o valor do lado: ");
    scanf_s("%f", &lado);

    volume = lado * lado * lado;

    printf("\nVolume da caixa: %f \n\n", volume);
}

//Declaração da função piscina
void piscina()
{
    float comprimento, largura, altura, volume;

    printf("Volume da piscina \n\n");
    printf("Digite o valor do comprimento: ");
    scanf_s("%f", &comprimento);
    printf("Digite o valor da largura: ");
    scanf_s("%f", &largura);
    printf("Digite o valor da altura: ");
    scanf_s("%f", &altura);

    volume = comprimento * largura * altura;

    printf("\nVolume da piscina: %f \n\n", volume);
}

```

7.

As variáveis locais são aquelas que são declaradas dentro de um bloco de comandos, delimitados pelas chaves. Este bloco pode ser uma função ou um bloco qualquer dentro de um programa.

As variáveis globais estarão acessíveis em todo o programa. A declaração das mesmas deve ser realizada fora de um bloco e geralmente são declaradas depois dos “includes” e antes da função “main”.

8.

As variáveis que são padrão em C/C++ são as do tipo Auto.

9.

As variáveis que retém os valores de uma chamada para outra são as do tipo Static.

10.

Nas três chamadas, surgirão valores iguais, pois são variáveis do tipo Auto, e os valores serão criados a cada chamada, portanto aparecendo nas três chamadas:

$$10 + 20 = 30$$

$$10 + 20 = 30$$

$$10 + 20 = 30$$

Na questão, quanto declaramos as variáveis como do tipo static, os valores irão permanecer de uma chamada para outra, portanto teremos como resposta:

$$1^{\text{a}} \text{ Vez:} \quad 10 + 20 = 30$$

$$2^{\text{a}} \text{ vez:} \quad 10 + 20 = 60$$

$$3^{\text{a}} \text{ vez:} \quad 10 + 20 = 90$$

Obs.: Repare que no código fonte os valores das variáveis: “valorA” e “valorB” não sofrem alterações, apenas a variável: “resultado” sofre alteração, pois está envolvida no cálculo:

resultado = resultado + valorA + valorB;

Razão pela qual parece que há erro na soma, mas como o valor e de resultado permanece ele utiliza o valor atualizado para o cálculo.

## 11.

```

// Capitulo01014_Exercicio01.cpp : Defines the entry point for the
console application.
//
#include "stdafx.h"
#include "stdlib.h"

float calcMedia(void);
int _tmain(int argc, _TCHAR* argv[])
{
    float media;

    for (int i = 0; i < 45; i++)
    {
        system("cls");
        printf("Digite o numero de matricula do aluno %d: ", i + 1);
        media = calcMedia();
        printf("Conforme sua media: %f , voce foi: ", media);
        if (media >= 6)
        {
            printf("APROVADO \n");
        }
        else
        {
            printf("REPROVADO \n");
        }
        system("PAUSE");
    }

    return 0;
}

float calcMedia(void)
{
    float n1, n2, n3, n4, media;
    int matricula;

    scanf_s("%d", &matricula);
    printf("Digite a primeira nota: ");
    scanf_s("%f", &n1);
    printf("Digite a segunda nota: ");
    scanf_s("%f", &n2);
    printf("Digite a terceira nota: ");
    scanf_s("%f", &n3);
    printf("Digite a quarta nota: ");
    scanf_s("%f", &n4);
    media = (n1 + n2 + n3 + n4) / 4;
    printf("\n\nA media do aluno %d foi: %f \n", matricula, media);

    return(media);
}

```

**12.**

```
// Capitulo01014_Exercicio02.cpp : Defines the entry point for the
console application.
//

#include "stdafx.h"
#include "stdlib.h"

int calcFatorial(int nro);

int _tmain(int argc, _TCHAR* argv[])
{
    int numero, resultado;

    printf("Digite um numero: ");
    scanf_s("%d", &numero);

    resultado = calcFatorial(numero);

    printf("Resultado do fatorial de %d foi: %d \n", numero, resultado);

    system("PAUSE");
    return 0;
}

int calcFatorial(int nro)
{
    int result=1;

    for (int i = 1; i <= nro; i++)
    {
        result *= i;
    }

    return(result);
}
```

**13.**

```
// Capitulo01014_Exercicio03.cpp : Defines the entry point for the
console application.
//

#include "stdafx.h"
#include "stdlib.h"

void calcPotencia(int base, int expoente);

int _tmain(int argc, _TCHAR* argv[])
```

```

{
    int valorBase, valorExpoente;

    printf("Digite o valor da base: ");
    scanf_s("%d", &valorBase);
    printf("Digite o valor do expoente: ");
    scanf_s("%d", &valorExpoente);

    calcPotencia(valorBase, valorExpoente);

    system("PAUSE");
    return 0;
}

void calcPotencia(int base, int expoente)
{
    int result = 1;

    for (int i = 1; i <= expoente; i++)
    {
        result *= base;
    }

    printf("Resultado de %d elevado ao expoente %d foi %d \n", base,
    expoente, result);
}

```

**14.**

```

// Capitulo01014_Exercicio04.cpp : Defines the entry point for the
// console application.
//

#include "stdafx.h"
#include "stdlib.h"

int calcSomatorio(int nro);

int _tmain(int argc, _TCHAR* argv[])
{
    int numero, resultado;

    printf("Digite o valor do numero: ");
    scanf_s("%d", &numero);

    resultado = calcSomatorio(numero);

    printf("O Somatorio dos numeros ate: %d foi %d \n", numero,
    resultado);

    system("PAUSE");
}

```

```
    return 0;
}

int calcSomatorio(int nro)
{
    int result = 0;

    for (int i = 1; i <= nro; i++)
    {
        result += i;
    }

    return(result);
}
```

---

## Capítulo 11

---

### 1.

Os valores das variáveis nas linhas identificadas são:

#### Item 1:

```
ValorA => 35
ValorB => 55
valor C => 55
```

#### Item 2:

```
ValorA => 35
ValorB => 90
valor C => 90
```

#### Item 3:

```
ValorA => 35
ValorB => 90
valor C => 35
```

#### Item 4:

```
ValorA => 35
ValorB => 90
valor C => 135
```

## 2.

Declarando uma variável do tipo ponteiro para um número real:

```
float *apontador;
```

Variável do tipo ponteiro recebendo o endereço de outra variável:

```
apontador = &outraVariavel;
```

Variável do tipo ponteiro, armazenando um valor, na variável para quem ele aponta:

```
*apontador = 77;
```

Somando um valor apontado por um ponteiro:

```
*apontador = *apontador + 1;
```

## 3.

```
// Capitulo0011_Exercicio03.cpp : Defines the entry point for the
// console application.
//
#include "stdafx.h"
#include "stdlib.h"

int _tmain(int argc, _TCHAR* argv[])
{
    int valorA, valorB;
    int *valorMaior;

    printf("Digite um numero: ");
    scanf_s("%d", &valorA);
    printf("Digite outro numero: ");
    scanf_s("%d", &valorB);

    if (valorA > valorB)
        valorMaior = &valorA;
    else
        valorMaior = &valorB;

    printf("Conteudo apontado: %d \n", *valorMaior);

    system("PAUSE");
    return 0;
}
```



---

## Capítulo 12

---

### 1.

```
// Capitulo0012_Exercicio01.cpp : Defines the entry point for the
console application.
//

#include "stdafx.h"
#include "stdlib.h"
#include "string.h"

int _tmain(int argc, _TCHAR* argv[])
{
    char frase[30];
    int tamanho;

    printf("Digite uma frase: ");
    gets_s(frase);

    tamanho = strlen(frase);

    printf("Tamanho da frase digitada foi: %d \n", tamanho);

    system("PAUSE");
    return 0;
}
```

### 2.

```
// Capitulo0012_Exercicio02.cpp : Defines the entry point for the
console application.
//

#include "stdafx.h"
#include "stdlib.h"
#include "string.h"

int _tmain(int argc, _TCHAR* argv[])
{
    char frase1[30], frase2[30], frase3[80] = "";
    int tamanho;

    printf("Digite uma frase: ");
    gets_s(frase1);
    printf("Digite outra frase: ");
    gets_s(frase2);

    printf("Frase 1 lida: %s \n", frase1);
}
```

```

tamanho = strlen(frase1);
printf("Tamanho da frase 1: %d \n", tamanho);

printf("Frase 2 lida: %s \n", frase2);
tamanho = strlen(frase2);
printf("Tamanho da frase 2: %d \n", tamanho);

strcat_s(frase3, frase1);
strcat_s(frase3, frase2);

printf("Frase 3 concatenada: %s \n", frase3);
tamanho = strlen(frase3);
printf("Tamanho da frase 3: %d \n", tamanho);

system("PAUSE");
return 0;
}

```

**3.**

```

// Capitulo0012_Exercicio03.cpp : Defines the entry point for the
console application.
//

#include "stdafx.h"
#include "stdlib.h"
#include "math.h"

int _tmain(int argc, _TCHAR* argv[])
{
    float resultado, base;

    printf("Digite o valor da base: ");
    scanf_s("%f", &base);

    for (int i = 0; i <= 10; i++)
    {
        resultado = pow(base, i);
        printf("%.0f elevado ao expoente %d foi: %.0f \n", base, i,
resultado);
    }

    system("PAUSE");
    return 0;
}

```

---

## Capítulo 13

---

1.

Não, pois um vetor permite apenas dados do mesmo tipo. Se forem dados de tipos diferentes e de informações diferentes não poderão ser armazenados em um vetor ou matriz.

2.

Sim, para guardar informações de tipos de dados diferentes e de informações diferentes a struct é uma forma apropriada.

3.

Nas linhas 1 e 2 devemos inserir as seguintes bibliotecas:

```
#include "stdio.h";  
#include "conio.h"
```

Na linha 3 para declarar a variável baseada na estrutura faremos:  
struct produto automóvel;

Na linha 4 o comando para limpar o buffer do teclado é:  
fflush(stdin);

4.

Código fonte do programa:

```
// Capitulo0013_Exercicio04.cpp : Defines the entry point for the  
// console application.  
//  
#include "stdafx.h"  
#include "stdlib.h"  
#include "stdio.h"  
#include "conio.h"  
  
struct cadastro //declaração do tipo da estrutura  
{  
    char nome[80];  
    float renda;  
    char email[80];  
    char fone[80];  
};
```

```

int _tmain(int argc, _TCHAR* argv[])
{
    struct cadastro cliente;    //declarando uma estrutura

    printf("Digite o nome do cliente: ");
    gets_s(cliente.nome);
    printf("Digite a renda familiar: ");
    scanf_s("%f", &cliente.renda);
    printf("Digite o email: ");
    fflush(stdin);
    gets_s(cliente.email);
    printf("Digite o telefone: ");
    gets_s(cliente.fone);

    printf("Nome do cliente.....: %s \n", cliente.nome);
    printf("Renda familiar.....: %.2f \n", cliente.renda);
    printf("Email.....: %s \n", cliente.email);
    printf("Fone.....: %s \n", cliente.fone);

    system("PAUSE");
    return 0;
}

```

## 5.

```

// Capitulo0013_Exercicio05.cpp : Defines the entry point for the
console application.
//

#include "stdafx.h"
#include "stdlib.h"
#include "stdio.h"
#include "conio.h"

struct cadastro
{
    char titulo[80];
    char autor[80];
    int paginas;
    float preco;
};

int _tmain(int argc, _TCHAR* argv[])
{
    struct cadastro livro;

    printf("Digite o titulo do livro: ");
    gets_s(livro.titulo);
    printf("Digite o autor: ");
    gets_s(livro.autor);
    printf("Digite a quantidade de paginas: ");

```

```

scanf_s("%d", &livro.paginas);
printf("Digite o preco: ");
scanf_s("%f", &livro.preco);

printf("Titulo do livro.....: %s \n", livro.titulo);
printf("Autor.....: %s \n", livro.autor);
printf("Paginas.....: %d \n", livro.paginas);
printf("Preco.....: %.2f \n", livro.preco);

system("PAUSE");
return 0;
}

```

---

## Capítulo 14

---

**1.**

O modo a ser utilizado será: `ios::trunc` que irá apagar o conteúdo existente no arquivo, se o mesmo existir.

**2.**

Para gravar uma série de números inteiros o modo ideal seria: `ios::binary` que irá gerar um arquivo em binário com os números armazenados.

**3.**

O código digitado na linha 1 para abrir o arquivo apagando o conteúdo anterior será:

```
arquivo.open("arquivo.txt", ios::trunc);
```

Na linha 2 o código para fechar o arquivo aberto será:

```
arquivo.close();
```

## 4.

```
// Capitulo0015_Exercicio04.cpp : Defines the entry point for the
console application.
//

#include "stdafx.h"
#include "stdlib.h"
#include <iostream>
#include <fstream>

using namespace std;

int _tmain(int argc, _TCHAR* argv[])
{
    char texto[80];

    ofstream arquivo;

    arquivo.open("arquivo.txt", ios::trunc);

    for (int i = 0; i < 10; i++)
    {
        cin.getline(texto, sizeof(texto));
        arquivo << texto << endl;
    }

    arquivo.close();

    system("PAUSE");
    return 0;
}
```

## 5.

```
// Capitulo0015_Exercicio05.cpp : Defines the entry point for the
console application.
//

#include "stdafx.h"
#include "stdlib.h"
#include <iostream>
#include <fstream>

using namespace std;

int _tmain(int argc, _TCHAR* argv[])
{
    char texto[80];
```

```
ifstream arquivo("arquivo.txt");

while (!arquivo.eof())
{
    arquivo.getline(texto, 80);
    cout << texto << endl;
}
arquivo.close();

system("PAUSE");
return 0;
}
```