



UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO – UFRPE
Bacharelado em Sistemas de Informação
Introdução à Computação

Roteiro para Aulas Práticas – C++

**Observação: Este material foi adaptado do material preparado para aulas práticas da disciplina Computação II
- PUC/Minas - Professora Cristiane Néri Nobre**

Maio de 2009

-
1. **abs (x)** - retorna o valor absoluto de x (x é inteiro)
 2. **ceil (x)** - retorna o menor inteiro não menor que x (x é real)
 3. **cos (x)** - retorna o cosseno de x (x, em radianos, é real)
 4. **exp (x)** - retorna o valor de e^x (x é real)
 5. **fabs (x)** - retorna o valor absoluto de x (x é real)
 6. **floor (x)** - retorna o maior inteiro não maior que x (x é real)
 7. **log (x)** - retorna o logaritmo neperiano de x (x é real)
 8. **log10 (x)** - retorna o logaritmo decimal de x (x é real)
 9. **pow (x,y)** - retorna o valor de x^y (x e y são reais)
 10. **sin (x)** - retorna o seno de x (x, em radianos, é real)
 11. **sqrt (x)** - retorna a raiz quadrada de x (x é real)
 12. **tan (x)** - retorna a tangente de x (x, em radianos, é real)

1.8 Algumas constantes matemáticas em C++

`<math.h>`

M_PI = 3.141593 (número π)
M_E = 2.718282 (número e)

1.9 Geração de números aleatórios em C++

`<stdlib.h>`

1. **randomize ()** inicia o gerador de números aleatórios
2. **random (x)**
 - a. retorna um número inteiro aleatório no intervalo [0, x-1], onde x é um número inteiro
3. **rand ()**
 - a. retorna um número inteiro aleatório no intervalo [0, RAND_MAX]. **RAND_MAX** é o maior inteiro definido.
4. **rand () % 100**
 - a. retorna um número inteiro aleatório no intervalo [0, 99]
5. **random (1000) + 1000**
 - a. retorna um número inteiro aleatório no intervalo [1000, 1999]

1.10 Manipuladores de tamanho de campos na saída em C++

1) Definidos na biblioteca: “**iomanip.h**” (`#include <iomanip.h>`)

2) Os manipuladores seguintes devem ser usados no “**cout**”:

- a) **setw** – seleciona o tamanho do próximo campo na saída. Os campos são alinhados à direita;
- b) **setprecision** – define o numero de casas decimais de um número “float” e arredonda;
- c) **setfill** – seleciona o caractere de preenchimento dos espaços em branco.

1.11 Atenção para erros mais freqüentes em C++

1. Ausência de ; no final de uma instrução. Normalmente o erro aparece na instrução seguinte.
2. Comando *include* ausente, nesse caso a mensagem de erro denuncia “falta de protótipo”.
3. Parêntesis não balanceados, isto é, o nº. de (deve ser igual ao nº. de) .
4. Ausência de **aspas** “” no final de uma string de caracteres.
5. Desbalanceamento de chaves e número desigual delas { } .
6. **ERRADO:** `cout<<"Resultado: ", hora, ":", min, ":", seg`
CORRETO: `cout << "Resultado : " << hora << " : " << min << " : " << seg;`
7. **ERRADO:** `cin >> A >> B >> C;`
CORRETO: `cin >> A >> B >> C;`
8. **ERRADO:** `if exp. Lógica;` **CORRETO:**
`if (exp. Lógica) → com parêntesis () e sem ponto-e-vírgula ;`
9. Uso indevido dos operadores = e == (p/ atribuição use = , p/ verificar igualdade use ==).
10. Não declaração de variável - mensagem: “undefined symbol”.
11. Escrever comando em maiúsculo. Todos os comandos (palavras-chave) de C++ devem ser escritos em letras minúsculas.
12. **ERRADO:** `int : A ;` **CORRETO:** `int A ; →`
sem dois-pontos : .
13. Falta de chaves no comando *if* quando existe mais de uma instrução a ser executada. Ex.: **CORRETO:** `if (exp. Lógica) { c;;; }`
14. Em C++, letras maiúsculas e minúsculas são diferentes.

1.12 Executando e compilando o seu programa:

- Para executar o seu programa clique em Ctrl+F9 ou no menu Run / Run
- Para salvar o seu programa clique no menu File / Save as

ROTEIRO 1 – COMANDOS DE SAÍDA E DE ENTRADA: COUT E CIN

1 - cout (pronuncia-se “C out) é um objeto de uma classe de I/O (leitura e impressão) predefinida em C++.

Sintaxe:

```
cout << “Teste”;
```

//Escreve a palavra teste na tela

```
cout << var;
```

// Escreve o valor armazenado na variável var

2 – O objeto cin (pronuncia-se “C out) manipula toda entrada do teclado por meio do operador de extração >> que conecta a entrada de dados à variável que a conterà.

As definições necessárias ao uso de cout e << e cin e >> estão no arquivo “iostream.h”.

Exercício 1

```
#include <conio.h> // bibliotecas de funções auxiliares
#include <iostream.h>

void main()
{
    cout<< "EXEMPLO 1 - Primeiro Programa em C++";
    getch();
}
```

Exercício 2

Altere o programa em C++ para:

```
#include <conio.h> // bibliotecas de funções auxiliares
#include <iostream.h>

void main() /* comentário delimitado */
{
    clrscr(); // função que limpa a tela
    cout<< "EXEMPLO 1 - Primeiro Programa em C++";
    getch(); // função que espera que uma tecla seja apertada
}
```

Exercício 3

Altere novamente o programa em C++:

```
#include <conio.h> // bibliotecas de funções auxiliares
#include <iostream.h>

void main() /* comentário delimitado */
{
    clrscr(); // função que limpa a tela
    cout<< “EXEMPLO 1 - Primeiro Programa em C++”;
    cout<< “\n”; // salta “n” linha

    cout << “Aluno: _____”;
    getch();
}
```

Exercício 4

Altere novamente o programa em C++:

```
#include <conio.h> // bibliotecas de funções auxiliares
#include <iostream.h>

void main() /* comentário delimitado */
{
    clrscr();
    cout<< "EXEMPLO 1 - Primeiro Programa em C++" <<"\n";
    cout << "\nAluno: _____" <<"\n";
    cout <<"\n";
    cout << "Exemplos de valores:";
    cout << "\nInteiro: " << 10 <<"\n";
    cout << "\nReal: " <<10.01 <<"\n";
    cout <<"\n";
    cout << "Tecla <enter> para terminar.";
    getch(); // espera que uma tecla seja apertada
}
```

Exercício 5

Digite e execute o programa abaixo:

```
#include <conio.h>
#include <iostream.h>

void main ()
{
    //Programa para ler um valor inteiro do teclado
    int x;
    cout <<"\nDigite o valor de x: ";
    cin >>x;
    cout <<"\nO resultado da leitura e: " <<x;
    getch();
}
```

Exercício 6

Faça um programa para:

- Ler e imprimir um número real, um caracter e uma string do teclado;

Exercício 7

Digite e execute o programa abaixo:

```
#include <conio.h>
#include <iostream.h>

void main ()
{
    //Programa para ler e operar com valores lidos do teclado
    int x, y;
    float result;
    cout <<"\nDigite o valor de x: ";
    cin >>x;
    cout <<"\nDigite o valor de y: ";
    cin >>y;
    result = (2*x) + (Y*Y);
    cout <<"\nO resultado da expressao e: " <<result;
    getch();
}
```

Exercício 8

Digite e execute o programa abaixo:

```
#include <conio.h>
#include <iostream.h>
void main ()
{
    //Programa para calcular velocidade
    //Variaveis:
    float D, T, V;          // Distancia, Tempo e Velocidade respectivamente
    clrscr();
    cout << "\nExemplo para calcular velocidade ";
    cout << "\nDigite uma distancia qualquer em [m]: ";
    cin >> D;
    cout << "\nDigite o tempo para percorre-la em [s]: ";
    cin >> T;
    V = D / T;
    cout << "\n V = D / T = " << V << "m/s";
    cout << "\nTecler <Enter> para terminar";
    getch();
    return;
}
```

Exercício 9

Faça um programa para:

- Ler a base e altura de um triângulo qualquer;
- Calcular e imprimir a área.
- Obs: $A = (b * h) / 2$;

Exercício 10

Repita o processo para calcular a área de um quadrado

Exercício 11

Faça um programa para:

- Ler os nomes e as notas de cinco alunos
- Calcular e imprimir a média aritmética das notas.

Exercício 12

Escreva um programa que solicite ao usuário a altura e o raio de um cilindro circular e imprima o volume do cilindro. O volume de um cilindro circular é calculado por meio da seguinte fórmula:

$$\text{Vol} = 3.141592 * \text{raio} * \text{raio} * \text{altura}$$

Exercício 13

Num cercado, há vários patos e coelhos. Escreva um programa que solicite ao usuário o total de cabeças e o total de pés e determine quantos patos e quantos coelhos encontram-se nesse cercado.

Exercício 14

Uma firma contrata um encanador a R\$40,00 por dia. Crie um programa que solicite o número de dias trabalhadas pelo encanador e imprima a quantia líquida que deverá ser paga, sabendo-se que são descontados 8% para imposto de renda

Exercício 15

A importância de R\$500.000,00 será dividida entre os três primeiros colocados de um concurso, em partes diretamente proporcionais aos pontos conseguidos por eles. Construa um programa que solicite o número de pontos dos três primeiros colocados e imprima a importância que caberá a cada um.

ROTEIRO 2 – COMANDO CONDICIONAL IF e IF - ELSE

O Comando IF

A instrução **IF** consiste na palavra-chave **IF** seguida de uma expressão de teste entre parênteses. Se a expressão de teste for verdadeira, a instrução é executada; do contrário, nada é feito.

O corpo de um **IF** pode ter uma única instrução terminada por pontos e vírgula ou várias instruções entre chaves.

Sintaxe:

```
if (expressão de teste)
{
    instrução;
    instrução;
}
```

Exercício 1 – Exemplo

Exercício 0 – Exemplo

```
#include <iostream.h>
#include <conio.h>

void main()
{
    cout << "\n Quantos anos você tem? ";
    int anos;
    cin >> anos;

    if (anos <30)
        cout <<" \n Você é muito jovem";

    getch();
}
```

Exercício 1

Faça um programa para:

- ler um valor inteiro e verificar se é par
- Se for par imprima uma mensagem

Exercício 2

Faça um programa que receba três números e verifique qual é o maior.

Exercício 3

Faça um programa que receba três números em ordem crescente e um quarto número que não siga esta regra. Mostre os quatro números em ordem decrescente.

O comando IF-ELSE

O comando **if-else** é uma expansão de um simples comando if. O comando if permite que executemos algo somente se a sua expressão de teste for verdadeira, caso contrário nada é executado.

Suponhamos que você queira executar alguma coisa se a expressão de teste for verdadeira e outra coisa se a expressão de teste for falsa. Neste caso, você deve usar o comando **if-else**.

Sintaxe:

A instrução if-else consiste num if seguido de uma instrução ou de um bloco de instruções, seguido da palavra-chave else, seguido de uma instrução ou um bloco de instruções.

Uma única instrução não necessita de chaves:

```
if (expressão de teste)
    instrução;
else
    instrução;
```

Várias instruções necessitam estar entre chaves:

```
if (expressão de teste)
{
    instrução;
    instrução;
}
else
{
    instrução;
    instrução;
}
```

Exercício 4 - Exemplo do comando if-else

```
#include <conio.h>
#include <iostream.h>

void main (){
    //Programa para ler e um inteiro e verificar se é zero
    int x;
    clrscr();
    cout <<"\nDigite o valor de x: ";
    cin >>x;
    if (x == 0)
        cout << "\nO valor digitado é zero";
    else
        cout << "\nO valor digitado não foi zero";
    cout << "\n DIGITE <ENTER> PARA TERMINAR";
    getch();
    return;
}
```

Exercício 5

Faça um programa para:

- Ler dois valores reais e verificar se são iguais
- Imprimir mensagem conforme o caso

Exercício 6

Faça um programa para calcular as raízes de uma equação de 2 grau. Os coeficientes da equação (A, B, C) serão informados pelo usuário. Verifique se o delta é negativo. Se for, mande a seguinte mensagem para o usuário: " Não existem raízes reais". Caso contrário envie as raízes.

Exercício 7

Faça um programa para :

- ler um valor e verificar se é par ou ímpar
- imprimir mensagem conforme o caso

Exercício 8

Faça um programa que escreva o seguinte menu na tela:

Este programa calcula as áreas das seguintes figuras geométricas:

- (1) Área do quadrado
- (2) Área do retângulo
- (3) Área do triângulo
- (4) Área trapézio

Baseado na escolha do usuário, você deverá pedir os valores necessários para fazer os cálculos e imprimir o valor da área para o usuário.

ROTEIRO 3 - switch

O comando **switch** permite selecionar uma entre várias ações alternativas. Embora construções **if-else** possam executar testes para escolha de uma entre várias alternativas, muitas vezes são deselegantes. O comando **switch** tem um formato limpo e claro.

A instrução **switch** consiste na palavra-chave **switch** seguida do nome de uma variável ou de um número constante entre parênteses. O corpo do comando **switch** é composto de vários casos rotulados por uma constante e opcionalmente um caso default. A expressão entre parênteses após a palavra-chave **switch** determina para qual caso será desviado o controle do programa.

O corpo de cada caso é composto por qualquer número de instruções. Geralmente, a última instrução é **break**. O comando **break** causa a saída imediata de todo o corpo do **switch**. Na falta do comando **break**, todas as instruções após o caso escolhido serão executadas, mesmo as que pertencem aos casos seguintes.

O comando **break** em somente dois usos em C++: em laços ou no comando **switch**.

Sintaxe:

```
switch (variável ou constante) //sem ponto e vírgula
{
case constante 1: //dois pontos
    instrução;
    instrução;
    break;
case constante 2: //dois pontos
    instrução;
    instrução;
    break;
default:
    instrução;
    instrução;
}

```

O corpo de um **switch** deve estar entre chaves.

Se o rótulo de um caso for igual ao valor da expressão do **switch**, a execução começa nele. Se nenhum caso for satisfeito e existir um caso default, a execução começará nele. Um caso default é opcional.

Não pode haver casos com rótulos iguais.

Exercício 1: Digite e execute o programa abaixo:

```
#include <conio.h>
#include <iostream.h>

void main (){
    //Programa para ler e testar um valor
    //Variável:
    int VALOR;
    clrscr();
    cout <<“\n EXEMPLO PARA LER E TESTAR UM VALOR””;
    cout <<“\n Entre com um valor: ”;
    cin >> VALOR;
    switch(VALOR)
    {
        case 0: cout <<“\n Foi digitado o ZERO””; break;
    }
}

```

```

        case 1: cout << "\n Foi digitado o valor UM"; break;
        case -1: cout << "\n Foi digitado o valor MENOS UM"; break;
        default : cout << "\n Foi digitado outro valor diferente de zero, um e menos um";
    }
    cout << "\n Tecla <ENTER> para terminar";
    getch();
    return;
}

```

Exercício 2 – Teste o programa abaixo

```

#include <conio.h>
#include <iostream.h>

void main ()
{
float n1,n2;
char op;

while(TRUE)          //sempre verdadeiro
{
cout << "\n Digite número operador número: ";
cin >> n1 >> op >> n2;
switch(op)
{
case '+': cout << n1+n2;break;
case '-': cout << n1-n2;break;
case '*': cout << n1*n2;break;
case '/': cout << n1/n2;break;
default: cout << "Operador desconhecido";
} //fim do switch

} //fim do while
} //fim do main

```

Exercício 3 – Utilize o comando switch para fazer um programa que calcula as áreas das seguintes figuras geométricas:

- (5) Área do quadrado
- (6) Área do retângulo
- (7) Área do triângulo
- (8) Área trapézio

Baseado na escolha do usuário, você deverá pedir os valores necessários para fazer os cálculos e imprimir o valor da área para o usuário.

Exercício 4

Utilize o comando switch e faça um programa para:

- Ler três valores inteiros
- Calcular a raiz quadrada da soma desses três valores
- Calcular o módulo da diferença entre os três valores
- Calcular a potência de cada número por 2, 4 e 8 respectivamente.
- Imprima os resultados.

Obs: Este resultados não aparecerão todos de uma vez para o usuário. Você deverá pedir os três valores ao usuário e este deverá escolher o que irá querer, conforme o menu abaixo:

- (1) Calcular a raiz quadrada da soma desses três valores
- (2) Calcular o módulo da diferença entre os três valores
- (3) Calcular a potência de cada número por 2, 4 e 8 respectivamente.

ROTEIRO 4 – COMANDOS DE REPETIÇÃO (LAÇOS)

Laços são comandos da linguagem C++ úteis sempre que uma ou mais instruções devam ser repetidas enquanto uma certa condição estiver sendo satisfeita.

Em C++ existem três estruturas de laços:

for
while
do-while

→ O laço for

O laço **for** é geralmente usado quando queremos repetir algo um número fixo de vezes. Isto significa que utilizamos um laço **for** quando sabemos de antemão o número de vezes a repetir.

O exemplo seguinte imprime uma linha de 20 '*' utilizando um laço **for** na sua forma mais simples.

```
#include <iostream.h>
#include <conio.h>
void main()
{
int i;
for (i=0; i<20; i++)
    cout << '*';
getch();
}
```

Sintaxe do laço for

A instrução **for** consiste na palavra-chave for seguida de parênteses que contêm três expressões separadas por pontos-e-vírgulas. Chamaremos a primeira destas expressões de **inicialização**, a segunda de **teste** e a terceira de **incremento**.

Exercício 1 - Digite e execute o programa abaixo:

```
#include <conio.h>
#include <iostream.h>

void main (void)
{
    //Programa para ler e N valores inteiros
    //Variáveis:
    int N, Valor, Contador;
    clrscr();
    cout << "\n pEXEMPLO PARA LER N VALORES;
    cout << "\n Entre com o valor de N: ";
    cin >> N;

    for (Contador = 1; Contador <= N; Contador = Contador + 1)
    {
        cout << "\n Entre um valor inteiro: ";
        cin >> Valor;
    }
    cout << "\n Foram lidos " << N << "valores";
    cout << "\n Tecla <ENTER> para terminar";
}
```

```

    getch();
    return;
}

```

Exercício 2

Faça um algoritmo que verifique se 5 alunos passaram ou não de ano da seguinte forma:

- ler o nome e as 4 notas de cada aluno
- calcular a nota final que é a soma destas 4 notas
- imprimir o nome do aluno e a nota final
- imprimir mensagem conforme o a nota:
 - caso a nota final for maior ou igual a 60 imprimir a mensagem “Aprovado”
 - caso a nota final for < 60 e >= 45 imprimir a mensagem “Recuperação”
 - caso a nota final for < 45 imprimir a mensagem “Reprovado”
- calcule e imprima o número de aluno que perderam média
- calcule e imprima a média da turma

Exercício 3

Faça um algoritmo que gere e calcule a série abaixo:

$$L = 1 - \frac{2}{4} + \frac{3}{9} - \frac{4}{16} + \dots + \frac{N}{N^2} \quad (\text{Obs: N deverá ser lido})$$

Exercício 4

Faça um algoritmo para calcular e imprimir o fatorial de um valor inteiro positivo qualquer que deve ser lido:

$$\text{Obs: } N! = 1 * 2 * 3 * \dots * N$$

Exercício 5

Foi feita uma pesquisa do IBOPE para canais de TV. De cada residência foi fornecido o número do canal em que a TV estava ligada (4 / 5 / 7 / 12). Faça um programa que:

- leia um número 2000 dados
- calcule e imprima a porcentagem de cada canal em relação ao número de dados lidos.

Exemplo:

$$P4 = \text{Cont4} \times 100 / 2000$$

Onde:

P4 = Porcentagem de residências ligadas no canal 4

Cont4 = Número de residências ligadas no canal 4

Exercício 6

A conversão de graus Fahrenheit para Centígrados é obtido por:

$$C = 5 (F - 32) / 9$$

Faça um programa que calcule e imprima uma tabela de graus Centígrados em função de graus Fahrenheit, que variem de 50 a 150 de um em um.

Saída da tela:

Graus Fahrenheit	Graus Centígrados
50	10
51	10.5556
.	.

→ O laço while

A segunda estrutura de laço em C++ é o laço **while**. À primeira vista, o laço **while** parece simples se comparado ao laço **for**; utiliza os mesmos elementos, mas eles são distribuídos de maneira diferente no programa.

Um laço **while** é apropriado em situações em que o laço pode ser terminado inesperadamente, por condições desenvolvidas dentro do laço.

Sintaxe do laço while

A instrução **while** consiste na palavra-chave **while** seguida de uma expressão de teste entre parênteses. Se a expressão de teste for verdadeira, o laço **while** é executado uma vez e a expressão de teste é avaliada novamente. Este ciclo de teste e execução é repetido até que a expressão de teste se torne falsa (igual a zero), então o laço termina e o controle do programa passa para a linha seguinte ao laço.

O corpo de um **while** pode ter uma única instrução terminada por ponto e vírgula, várias instruções entre chaves ou ainda nenhuma instrução mantendo o ponto e vírgula.

Exercício 1 - Digite e execute o programa abaixo:

```
#include <conio.h>
#include <iostream.h>

void main ()
{
    //Programa para ler e diversos valores inteiros – flag é valor = 0

    //Variáveis:
    int X;
    clrscr();
    cout <<“\n pEXEMPLO PARA LER DIVERSOS VALORES;
    cout <<“\n Entre com o primeiro valor: ”;
    cin >> X;
    while (X != 0)
    {
        cout <<“\n Entre com outro valor: ”;
        cin >> X;
    }
    cout <<“\n Tecla <ENTER> para terminar”;
    getch();
    return;
}
```

Exercício 2

Acrescente um contador no programa anterior para contar o número de vezes em que foi lido um valor.

Exercício 3

Digite e execute o programa abaixo. Ele mostra como acrescentar uma rotina para que o programa seja repetido outras vezes sem precisar voltar ao editor de código.

```
#include <conio.h>
#include <iostream.h>

void main ()
{
```

```

//Programa para ler diversos valores
//Variáveis:
char Nome[15], Repetir;
Repetir = 'S';
while (Repetir == 'S' || Repetir == 's')
{
    clrscr();
    cout << "\n EXEMPLO PARA LER DIVERSOS NOMES";
    cout << "\n Entre com o primeiro nome: ";
    cin >> Nome;
    while (Nome[0] != 'F' && Nome[1] != 'T' && Nome[3] != 'M')
    {
        cout << "\n Entre com outro nome: ";
        cin >> Nome;
    }
    cout << "\n Deseja repetir o programa ? (S / N) ";
    cin >> Repetir;
}
return;
}

```

Exercício 4

Acrescente uma rotina no exercício 2 que permita que o programa seja executado várias vezes.

Exercício 5

Faça programas que implementem os seguintes somatórios:

$$H = 1 + 1/2 + 1/3 + 1/4 + \dots + 1/n \quad \text{(Obs: N deverá ser lido)}$$

$$G = 1 + 3/2 + 5/3 + 7/4 + 99/50$$

Exercício 6

Faça um programa que:

- leia três valores
 - verifique se esses valores podem formar ou não um triângulo
- Obs: para formar triângulo as seguintes condições têm que ser verdadeiras:
- $$\begin{aligned} \text{Lado1} &< \text{Lado2} + \text{Lado3} \\ \text{Lado2} &< \text{Lado1} + \text{Lado3} \\ \text{Lado3} &< \text{Lado1} + \text{Lado2} \end{aligned}$$
- caso possam formar triângulo classificá-lo em isósceles, equilátero ou escaleno
 - caso não possam, imprimir mensagem: NÃO É TRIANGULO
 - acrescente uma rotina que permita que o programa seja executado outras vezes.

→ O laço do-while

A terceira e última estrutura de laço em C++ é o laço **do-while**. Este laço é bastante similar ao laço **while** e é utilizado em situações em que é necessário executar o corpo do laço uma primeira vez e depois avaliar a expressão de teste e criar um ciclo repetido.

Sintaxe do laço do-while

A instrução **do-while** consiste na palavra-chave **do** seguida de um bloco de uma ou mais instruções entre chaves e terminada pela palavra-chave **while** seguida de uma expressão de teste entre parênteses terminada por ponto e vírgula.

Primeiramente, o bloco de código é executado, em seguida a expressão de teste entre parênteses é avaliada; se verdadeira, o corpo do laço é mais uma vez executado e a expressão de teste é avaliada novamente. Este ciclo de execução do bloco e teste é repetido até que a expressão de teste se torne falsa (igual a zero), então o laço termina e o controle do programa passa para a linha seguinte do laço.

Exercício 1 - Digite e execute o algoritmo abaixo:

```
#include <conio.h>
#include<iostream.h>

void main(void)
{
    // Programa para ler e imprimir 03 inteiros
    // Variáveis:
    int X, Contador;
    clrscr();
    cout << "\n Exemplo para ler e imprimir 03 valores inteiros";
    Contador = 1;
    // Essa é a estrutura repita até – repetição com teste no final
    do
    {
        cout<<"\n"<<Contador<< " Digite um valor inteiro: ";
        cin>> X;
        cout<<"\n O valor digitado foi: "<<X ;
        Contador = Contador + 1;
    }
    while (Contador <=3);
    cout<<"\n Tecla <Enter> para terminar";
    getch();
    return;
}
```

Exercícios de Revisão – Estruturas de Controle (Condicional e Repetição)

Usando a(s) estrutura(s) de controle que considerar conveniente(s) faça os programas abaixo:

Exercício 2:

Em um cardápio de um restaurante existe os seguintes pratos do dia:

– PRATOS DO DIA –			
CÓDIGO	PRATO	CONTEÚDO	PREÇO
01	COCÓ	Arroz, Frango Frito, Feijão, Batata Sotê	R\$ 4,00
02	MUMU	Filé grelhado, Arroz à Grega, Salada Mista	R\$ 5,00
03	QUAQUÁ	Pato com Laranja, Arroz, Maionese	R\$ 6,50
04	SUPERCOCÓ	Strogonoff, Arroz, Fritas, Maionese	R\$ 6,00

BEBIDAS		
CÓDIGO	BEBIDA	PREÇO
11	Refrigerante em lata	R\$ 1,20
12	Sucos Naturais	R\$ 1,50
13	Água Mineral – Garrafa	R\$ 0,80
14	Cerveja - Garrafa	R\$ 1,80

SOBREMESAS		
CÓDIGO	TIPO	PREÇO
21	Salada de Frutas	R\$ 1,00

22	Doces	R\$ 0,70
23	Sorvete – bola	R\$ 0,80
24	Mousse	R\$ 1,20

Faça um programa que:

- leia o código do prato do dia escolhido pelo freguês
- leia o código da bebida escolhida pelo freguês
- leia o código da sobremesa escolhida pelo freguês
- calcule o valor da conta a pagar sabendo que o garçom terá 10%
- imprima o valor da conta.
- Calcule o valor arrecadado pelo caixa no final do dia (quando não se desejar fechar mais nenhuma conta)
- Acrescente rotinas que permitam ao cliente comer mais de um prato, beber mais de uma bebida e comer mais de uma sobremesa.

Observações:

1 – o freguês poderá ir ao restaurante e apenas comer o prato do dia e não beber nada, nem comer sobremesa. Neste caso, considere o referido código como 00.

2 – o freguês poderá comer apenas 1 prato, 1 bebida e 1 sobremesa

3 – você deverá repetir o fechamento da conta para vários clientes – flag repetir = 'N'

Você poderá continuar o programa a partir do seguinte trecho:

```
#include <conio.h>
```

```
#include <iostream.h>
```

```
void main(void)
```

```
{
```

```
float Preco, PP, PB, PS;
```

```
int CodPrato, CodBeb, CodSob;
```

```
char Repetir;
```

```
do
```

```
{
```

```
clrscr();
```

```
cout<<"\n FECHAMENTO DE CAIXA";
```

```
cout<<"\n Prato: ";
```

```
cin>> CodPrato;
```

```
switch (CodPrato)
```

```
{
```

```
case 1: PP = 4.00; break;
```

ROTEIRO 5 – Matrizes

→ Matriz com uma dimensão (vetores)

Matriz é um tipo de dado em C++ usado para representar uma coleção de variáveis de mesmo tipo.

Imagine o seguinte problema: Calcular a média aritmética das notas de prova de cinco alunos. Você poderia escrever o seguinte programa:

```
#include <iostream.h>
#include <conio.h>

void main()
{
int nota0, nota1, nota2, nota3, nota4;

cout << "Digite a nota do aluno 1"; cin >> nota0;
cout << "Digite a nota do aluno 2"; cin >> nota1;
cout << "Digite a nota do aluno 3"; cin >> nota2;
cout << "Digite a nota do aluno 4"; cin >> nota3;
cout << "Digite a nota do aluno 5"; cin >> nota4;

int media = (nota0+nota1+nota2+nota3+nota4)/5;
cout << "\n A média das notas é: " << media;
getch();
}
```

Imagine agora se você pretendesse encontrar a média aritmética das notas de uma classe de 50 alunos ou de uma escola toda com 2000 alunos? Seria uma tarefa bastante volumosa!

É evidente que precisamos de uma maneira conveniente para referenciar tais coleções de dados similares. **Matriz** é o tipo de dado oferecido por C++ para esse propósito.

Uma **matriz** é um conjunto de variáveis, do mesmo tipo, referenciadas por um único nome, onde cada variável é diferenciada por meio de um número chamado "índice". Os colchetes são usados para conter o índice.

A declaração

```
int notas[5];
```

aloca memória para armazenar cinco variáveis do tipo int e anuncia que notas é uma matriz de cinco elementos.

Vamos reescrever programa anterior usando uma matriz:

```
#include <iostream.h>
#include <conio.h>

void main()
{
int notas[5], media =0;

for (int i=0; i<5;i++)
{
cout << "Digite a nota do aluno " << (i+1) << ": ";
cin >> notas[i]
media +=notas[i];
}
cout << "\n A média das notas é: " << (media/5);
getch();
}
```

Exercício 1

Digite e execute o programa abaixo: (O programa lê e imprime um vetor de 10 posições).

```
#include <conio.h>
#include <iostream.h>
typedef int Vetor[10];
void main(void)
{
//Programa para ler um vetor de inteiros
//Variáveis
Vetor v;
int I;
clrscr();
cout<<"\n EXEMPLO PARA LER UM VETOR DE INTEIROS<< "\n";
for (I = 0; I <= 10; I++)
{
cout<<"\n Digitar o "<<I<<"º Inteiro: ";
cin>> v[I-1]; // a primeira posição é zero, por isso o índice I-1
}
cout<<"\n";
cout<<"\n Vetor Lido: ";
cout<<"\n";
for (I = 0; I <=10; I++)
{
cout<<v[I-1]<<" ";
}
cout<<"Tecla <ENTER> para terminar";
getch();
return;
}
```

Exercício 2

Em uma cidade, sabe-se que, de janeiro ao final de fevereiro de 2001, não ocorreu temperatura inferior a 15°C, nem superior a 40°C. Faça um programa que armazene as temperaturas de cada dia em um vetor (de 59 posições), calcule e imprima:

- a) A menor e a maior temperatura ocorrida
- b) A temperatura média
- c) O número de dias nos quais a temperatura foi inferior a temperatura média.
- d)

Exercício 3

Faça um programa que leia 10 números positivos, armazene-os em um vetor e imprima os números primos e as posições correspondentes no vetor.

Exercício 4

Faça um programa que leia 10 números em qualquer ordem, armazene-os em um vetor e imprima este vetor ordenado.

Exercício 5

Faça um programa para corrigir provas de múltipla escolha. Cada prova tem 10 questões valendo um ponto cada. O primeiro conjunto de dados a ser lido será o gabarito da prova. Os outros dados serão os números dos alunos e suas respectivas respostas, e o último número do aluno fictício será 9999. O programa deve calcular e imprimir:

- a) para cada aluno, o seu número e sua nota
- b) a porcentagem de aprovação, sabendo que a nota mínima de aprovação é 6.
- c) A nota que teve a maior frequência absoluta.

Exercício 6

Faça um programa que faça reservas de passagens aéreas de uma companhia. Além da leitura do número de vôos e quantidade de dados disponíveis, ler vários pedidos de reserva, constituído do numero da carteira de identidade do cliente e do número do vôo desejado.

Para cada cliente, verificar se há disponibilidade no vôo desejado. Em caso afirmativo imprimir o número da identidade do cliente e o número do vôo, atualizando o número de lugares disponíveis. Caso contrário, avisar o cliente da inexistência de lugares. Indicando o fim existe um passageiro cuja identidade é 9999. Considere que a companhia tenha 10 vôos.

→ Matrizes de mais de uma dimensão

Os elementos de uma matriz podem ser de qualquer tipo, incluindo matrizes. Os exemplos vistos até agora são de matrizes de uma dimensão, isto é, em que os elementos são de tipos simples. Diremos que uma matriz é de duas dimensões quando seus elementos forem matrizes de uma dimensão.

Com dois pares de colchetes obtemos uma matriz de duas dimensões e com cada par de colchetes adicionais obtemos matrizes com uma dimensão a mais.

Exercício 1

Digite e execute o programa abaixo:

(O programa lê e imprime uma matriz 3 x 3 de valores inteiros).

```
#include <conio.h>
#include <iostream.h>

typedef int Matriz[3][3];

void main(void)
```

```

{
//Programa para ler uma matriz de inteiros
//Variáveis
Matriz M;
int Linha , Coluna;
clrscr();
cout<<"\n EXEMPLO PARA LER UMA MATRIZ DE INTEIROS";
for (Linha = 0; Linha <= 2; Linha++)
{
    for ( Coluna = 0; Coluna <= 2; Coluna++)
    {
        cout<<"\n";
        cout<<"\n Digitar o elemento M["<<Linha<<","<<Coluna<<"]: ";
        cin>> M[Linha][Coluna];

    }
}
cout<<"\n";
cout<<"\n Matriz Lida: ";
cout<<"\n \n";
for (Linha = 0; Linha <= 2; Linha++)
{
    for(Coluna = 0; Coluna <=2; Coluna++)
        cout<<" " <<M[Linha][Coluna]<<" ";
    cout<<"\n";
}
cout<<"\n Tecla <ENTER> para terminar";
getch();
return;
}

```

Exercício 2

Faça um programa que :

- leia duas matrizes A e B de 2 dimensões (3x3) de elementos inteiros
- calcule e imprima a matriz C que é a soma vetorial das duas anteriores
- calcule e imprima a soma dos elementos da diagonal principal da matriz C
- calcule e imprima a matriz D que é a transposta de C
- calcule e imprima o produto vetorial de A x B

Exercício 3

Ler alguns números inteiros e positivos (“flag”: -999) e informar:

- Quantos números foram fornecidos (o “flag” não deve ser contado),
- Qual é o maior e qual é o menor número lido,
- Quantos números são pares e quantos são ímpares,
- A soma dos números lidos.

Exercício 4

Gerar aleatoriamente 200 números inteiros e positivos no intervalo [1, 9999] e informar:

- Qual é o maior e qual é o menor número gerado,
- Quantos números são pares e quantos são ímpares,
- A soma dos números gerados.

Exercício 5

Tem-se um conjunto de dados contendo a altura em metros e o sexo (‘F’ para feminino e ‘M’ para masculino) de 50 pessoas. Escreva um programa para ler esses dados e informar:

- A maior e a menor altura do grupo;
- A média das alturas das mulheres;
- O número de homens com altura superior a 2 metros.

Exercício 6

Ler as notas dos alunos de uma turma (não se sabe quantos alunos são, mas sabe-se que o “flag” é uma nota negativa) e informar:

- quantos alunos existem na turma;
- a média aritmética das notas da turma;
- quantos alunos foram aprovados (nota mínima para aprovação: 60).

Roteiro 6 – Funções

Uma função é um conjunto de instruções desenhadas para cumprir uma tarefa particular e agrupadas numa unidade com um nome para referenciá-la.

A principal razão para usar funções é a de dividir a tarefa original em pequenas tarefas que simplificam e organizam o programa como um todo.

Uma outra razão para escrever funções é a de reduzir o tamanho do programa. Qualquer seqüência de instruções que apareça no programa mais de uma vez é candidata a ser uma função.

O código de uma função é agregado ao programa uma única vez e pode ser executado muitas vezes no decorrer do programa.

Chamando uma função

Um programa pode conter uma ou mais funções, das quais uma delas deve ser **main()**. A execução do programa sempre começa em **main()** e, quando o controle do programa encontra uma instrução que inclui o nome de uma função, a função é chamada.

Chamar uma função é o meio pelo qual solicitamos que o programa desvie o controle e passe para função, execute suas instruções e depois volte à instrução seguinte à da chamada a função.

O protótipo de funções

Da mesma forma que com relação a uma variável, não podemos também usar uma função sem antes declará-la.

A declaração de uma função é chamada protótipo e é uma instrução, geralmente colocada no início do programa, que estabelece o tipo da função e os argumentos que ela recebe.

Veja o exemplo a seguir:

Exemplo:

```
#include<iostream.h>
void main()
{
    int celsius(int fahr);    //protótipo local
    int c, f;

    cout << "\n Digite a temperatura em graus Fahrenheit: ";
    cin > f;

    c = celsius(f);          //chamada à função
    cout << "\n Celsius = " << c;
}
//Celsius()
//Definição da função

int celsius(int fahr)
{
    int c;
    c=(fahr - 32) * 5/9;
    return c;
}
```

Eliminando o protótipo de funções

Se a definição for feita antes da instrução de sua chamada, o seu protótipo não é obrigatório.

Desta forma, o exemplo acima poderia ser reescrito da seguinte forma:

```
#include<iostream.h>

//Celsius()
//Definição da função

int celsius(int fahr)
{
    int c;
    c=(fahr - 32) * 5/9;
    return c;
}

void main()
{
    int c, f;

    cout << "\n Digite a temperatura em graus Fahrenheit: ";
    cin > f;

    c = celsius(f);          //chamada à função
    cout << "\n Celsius = " << c;
}
```

Exercício 1

Digite e execute o programa abaixo:

(Programa que mostra a chamada e execução de um procedimento sem parâmetros)

```
#include <conio.h>
#include <iostream.h>

void Procedimento(void)
{
    cout<<"\n";
    cout<<"\n Procedimento sem parametros ";
    cout<<"\n";
    return;
}

void main(void)

{
    //Programa executar um bloco chamado procedimento sem parâmetros
    clrscr();
    cout<<"Exemplo de chamada de procedimento";
    Procedimento();
    cout<<"\n Tecla <Enter> para terminar";
    getch();
    return;
}
```

Exercício 2

Faça um programa que:

- crie um procedimento chamado Mensagem_de_Erro, sem parâmetros, para imprimir a seguinte mensagem na tela:
"Erro codigo 001 – operação inválida"

Exercício 3

Digite e execute o programa abaixo:

(Programa que mostra a chamada e execução de um procedimento com parâmetros)

```
#include <conio.h>
#include <iostream.h>

void Soma(float a, float b)
{
    float c;
    cout<<"\n";
    cout<<"\n Procedimento que soma valores ";
    c = a + b;
    cout<<"\n O resultado da soma é "<< c;
    return;
}

void main(void)
{
    float x, y;
    //Programa executar um bloco chamado Soma com passagem de parâmetros
    clrscr();
    cout<<"Exemplo de chamada de procedimento com parâmentos";
    cout<<"\n Entre com o primeiro valor: ";
    cin>>x;
    cout<<"\n Entre com o segundo valor : ";
    cin>>y;
    Soma(x,y);
    cout<<"\n Tecele <Enter> para terminar";
    getch();
    return;
}
```

Exercício 4

Faça um programa que incremente o exercício anterior de tal forma:

- a) acrescente um procedimento para realizar uma subtração
- b) acrescente um procedimento para realizar uma divisão
- c) acrescente um procedimento para realizar uma multiplicação

Obs: você poderá usar a estrutura escolha conforme o menu abaixo no programa principal:

```

.
.
.
cout<<"\n [1] - SOMA";
cout<<"\n [2] - SUBTRAÇÃO";
cout<<"\n [3] - DIVISÃO";
cout<<"\n [4] - MULTIPLICAÇÃO";
cout<<"\n\n Opção: ";
cin>> opcao;
switch (opcao)
{
    case 1: Soma(x,y); break;
    case 2: Subtrai(x,y); break;
    case 3: Dividi(x,y); break;
    case 4: Multiplica(x,y); break;
    default : cout<< "\n OPCAO INVALIDA";
}
.
.

```

-
- d) depois do programa estar pronto faça uma rotina para que o programa possa ser repetido várias vezes
sugestão:
- ```
do
{
...
}
while (repetir == 'S') || (repetir == 's')
```

**Exercício 5**

Escreva uma função de nome *soma* ( ) que receba um número inteiro e positivo **N** como argumento e retorne a soma dos **N** primeiros números inteiros e positivos.

**Exercício 6**

Escreva uma função chamada *potência* ( ) que aceite dois argumentos inteiros e positivos *i* e *j*. A função deverá retornar o valor de *i* elevado à potência *j*. Lembre-se que uma potência é um produto de fatores iguais.

**Exercício 7**

Escreva uma função de protótipo: *int max ( int vet[ ], int tam);*  
que recebe um vetor de inteiros e seu tamanho e retorna o maior elemento dele.

**Exercício 8**

Escreva uma função de protótipo: *int min ( int vet[ ], int tam);*  
que recebe um vetor de inteiros e seu tamanho e retorna o menor elemento dele;

**Exercício 9**

Escreva uma função que recebe uma seqüência (vetor) de números inteiros e inverta a ordem de seus elementos.

### Mais exercícios propostos

Exemplo:

```
include <conio.h>
include <iostream.h>
include <iomanip.h>
void main()
{
int uva = 37, lar = 12345;
float arroz = 12.3456, feijao = 234.53;
clrscr();
cout<<"\nUvas: " <<setw(13)<< uva;
cout<<"\nLaranjas: " <<setw(13)<<lar;
cout << setfill('.');
cout<<"\nUvas: " <<setw(13)<<uva;
cout<<"\nLaranjas: " <<setw(13)<<lar;
cout << setfill(' ');
cout<<"\n'"<<setw(13)<<"Uvas:"<<setw(8)<<uva;
cout<<"\n'"<<setw(13)<<"Laranjas:"<<setw(8)<<
laranja;
cout << setprecision(2);
cout<<"\nArroz: " << setw(13) << arroz;
cout<<"\nFeijao: " << setw(13) << feijao;
getch ();
}
```

Saída obtida:

```
Uvas: 37
Laranjas: 12345
Uvas: 37
Laranjas: 12345
 Uvas: 37
 Laranjas: 12345
Arroz: 12.35
Feijao: 234.53
```

### MANIPULADORES DE BASES NUMÉRICAS NA SAÍDA EM C++

- 1) Definidos na biblioteca: “`iostream.h`” ( `#include <iostream.h>` )
- 2) Os manipuladores seguintes devem ser usados no “`cout`”:  
 dec – a impressão do próximo campo é apresentada em decimal (default);  
 hex – a impressão do próximo campo é apresentada em hexa-decimal  
 oct – a impressão do próximo campo é apresentada em octal

Exemplo:

```
include <conio.h>
include <iostream.h>
include <iomanip.h>
void main()
{ int num = 65;
 clrscr();
 cout << "\nHexadecimal: " <<setw(3)<<hex<< num;
 cout << "\nOctal: " <<setw(3)<<oct<< num;
 cout << "\nDecimal: " <<setw(3)<<dec<< num;
 cout << "\nCaractere: " <<setw(3)<<char(num);
 getch ();
}
```

Saída obtida:

---

Hexadecimal: 41  
Octal: 101  
Decimal: 65  
Caractere: A

POSICIONAMENTO DO CURSOR NA SAÍDA (TELA) EM C++

A instrução “gotoxy(c,l)” é usada para se posicionar o cursor antes de um “cout”, onde “c” é a coluna (de 1 a 80) e “l” a linha (de 1 a 25).

---

**EXERCÍCIOS**

1. Mostrar todos os números inteiros existentes no intervalo:  $[-20, +30]$
2. Mostrar todos os múltiplos de 7 menores que 1000. Informe, no final, quantos são.
3. Ler dois números inteiros A e B (sendo obrigatório  $A \leq B$ ). Listar todos os números inteiros que pertencem ao intervalo:  $[A, B]$ .
4. Ler dois números inteiros A e B (sendo obrigatório  $A \leq B$ ). Listar todos os números inteiros ímpares que pertencem ao intervalo:  $[A, B]$ .
5. Ler um número inteiro e positivo N e mostrar, em ordem decrescente, todos os números inteiros ímpares que pertencem ao intervalo:  $[N, N/2]$ .
6. Ler um número inteiro e positivo N e mostrar, em ordem decrescente, todos os números inteiros pares que pertencem ao intervalo:  $[N, N/2]$ .
7. Ler um número inteiro e positivo N e mostrar, em ordem decrescente, todos os divisores de N.
8. Ler um número inteiro e positivo N e informar a soma dos quadrados dos N primeiros números inteiros e positivos.
9. Ler dez números inteiros e mostrar os que forem divisíveis por 7.
10. Ler um número inteiro e positivo N e informar o N!
11. Calcular e mostrar uma tabela de raios e áreas de círculos com o raio variando de 1 a 20 cm de 1 em 1 cm.
12. Sabendo que 1 polegada = 2,54 cm, elaborar uma tabela de conversão de polegadas em centímetros no intervalo de 1 a 20 polegadas em incrementos de 0,2 polegada.
13. Elaborar uma tabela para conversão de temperaturas Fahrenheit em temperaturas Célsius, fazendo a temperatura Fahrenheit variar de 0 a 200 graus, em intervalos de 10 graus.  

$$T_C = (5/9)(T_F - 32)$$
14. Ler dez números reais e informar qual é o maior, qual é o menor e qual é a média aritmética dos mesmos.
15. Ler um número inteiro e positivo N e calcular a soma:  $S = 1 + 2 + 3 + \dots + N$ .
16. Ler um número inteiro e positivo N e um outro real X e calcular a soma:  $S = X + 2X + 3X + \dots + NX$ .
17. Calcular e informar qual é o N-ésimo (N deverá ser lido do teclado) termo da série:  

$$a. \quad 1 \quad 1 \quad 2 \quad 4 \quad 7 \quad 11 \quad 16 \quad 22 \quad \dots\dots$$
18. Ler um número inteiro e positivo qualquer "N" e mostrar na tela os "N" primeiros termos da série abaixo e a sua respectiva soma.

---

*1. 1 1 2 6 24 120 720 5040 .....*

19. Ler um número inteiro e positivo qualquer “ $N$ ” e mostrar na tela os “ $N$ ” primeiros termos da série abaixo e a sua respectiva soma.

*a. 1 1 2 3 4 9 8 27 16 81 32 243  
.....*

20. Sendo dados de uma progressão aritmética: o termo inicial 2 e razão 3, ler do teclado um número inteiro  $N$  e informar o  $N$ -ésimo termo desta p.a. .

21. Ler um número inteiro e positivo  $N$ . Calcular, com  $N$  parcelas, o somatório:

$$S = \frac{1}{2}^0 + \frac{1}{2}^1 + \frac{1}{2}^2 + \dots$$

Ao executar, observar o limite do somatório à medida que  $N$  cresce.

22. Ler um número inteiro  $N$  e mostrar todos os termos da seqüência de Fibonacci menores que  $N$ .

23. Ler alguns números inteiros e positivos (“flag”: -9999) e informar:

- Quantos números foram fornecidos (o “flag” não deve ser contado),
- Qual é o maior e qual é o menor número lido,
- Quantos números são pares e quantos são ímpares,
- A soma dos números lidos.

24. Gerar aleatoriamente 200 números inteiros e positivos no intervalo [1, 9999] e informar:

- Qual é o maior e qual é o menor número gerado,
- Quantos números são pares e quantos são ímpares,
- A soma dos números gerados.

25. Tem-se um conjunto de dados contendo a altura em metros e o sexo ( ‘F’ para feminino e ‘M’ para masculino) de 50 pessoas. Escreva um programa para ler esses dados e informar:

- A maior e a menor altura do grupo;
- A média das alturas das mulheres;
- O número de homens com altura superior a 2 metros.

26. Ler as notas dos alunos de uma turma (não se sabe quantos alunos são, mas sabe-se que o “flag” é uma nota negativa) e informar:

- quantos alunos existem na turma;
- a média aritmética das notas da turma;
- quantos alunos foram aprovados (nota mínima para aprovação: 60).



27. Calcular, com 20 parcelas (sendo  $x$  a medida de um arco em radianos):

$$\text{sen}(x) = x - x^3/3! + x^5/5! - x^7/7! + x^9/9! \dots$$

28. Por subtrações sucessivas pode-se calcular o MDC de dois números inteiros usando as propriedades abaixo:

$$\text{MDC}(x, y) = x \quad \text{se } x = y$$

$$\text{i. } = \text{MDC}(x - y, y) \quad \text{se } x > y$$

$$\text{ii. } = \text{MDC}(x, y - x) \quad \text{se } x < y$$

29. Calcular o MDC de dois números inteiros (recebidos através do dispositivo de entrada) pelo método das subtrações sucessivas descrito acima.

30. Sabendo que  $A * B = \text{MDC}(A, B) * \text{MMC}(A, B)$ , calcule o MMC de dois números inteiros lidos do teclado, usando o método das subtrações sucessivas para calcular o MDC deles.

31. Ler um número inteiro e positivo  $N$  e escrever seus algarismos de trás para frente.

32. Gerar uma tabuada da multiplicação de fatores contendo apenas um algarismo ( $0 \times 0$  até  $9 \times 9$ ).

33. Ler um número inteiro  $N$  e calcular o seguinte somatório:

$$34. \quad i=1 \qquad i=2 \qquad i=3 \qquad i=N$$

$$35. \quad \sum_{i=1}^N i = \sum_{i=1}^1 i + \sum_{i=1}^2 i + \sum_{i=1}^3 i + \dots + \sum_{i=1}^N i$$

36. Usando a função random ( $X$ ) que gera um número inteiro aleatório entre 0 e  $X - 1$ , escreva um programa para gerar um número aleatório secreto  $S$  (entre 1 e 1000) e solicitar que seja digitado um número  $N$  (de 1 a 1000) no teclado. Se  $N = S$ , informar “Você acertou” e encerrar o programa. Agora, se  $N > S$ , informar “Número Grande” ou se  $N < S$  informar “Número Pequeno” e solicitar novo número pelo teclado informando o novo intervalo possível.

37. Ler um número indeterminado de linhas contendo cada uma a idade de um indivíduo. A última linha, que não entrará nos cálculos, contém o valor de idade igual a zero (flag). Calcular e mostrar a idade média desse grupo de indivíduos

38. Ler a altura e o sexo de um grupo de 50 pessoas. Calcular e mostrar:

- A maior e a menor altura do grupo;
- A média de altura das mulheres;
- O número de homens.

39. Supondo que a população de um país A seja 90.000.000 (noventa milhões) de habitantes, com uma taxa anual de crescimento de 3 % e que a população de um outro país B seja de 200.000.000 (duzentos milhões) de habitantes, com uma taxa anual de crescimento de 1,5%, fazer um programa que calcule e escreva o número de anos necessários para que a população do país A ultrapasse ou iguale a população de B, mantidas essas taxas de crescimento.
40. Um determinado material radioativo perde metade de sua massa a cada 50 segundos. Dada a massa inicial, em gramas, escrever um programa para determinar o tempo necessário para que essa massa se torne menor que 0,5 grama. Escreva a massa inicial, a massa final e o tempo calculado em horas, minutos e segundos.
41. Ler um número real R e calcular e escrever a sua raiz quadrada e a sua raiz cúbica. Use o método das aproximações sucessivas até encontrar um erro menor que 0,00001. Usar as fórmulas:
- Para a raiz quadrada:  $X_{i+1} = (X_i + R / X_i) / 2$
  - Para a raiz cúbica:  $X_{i+1} = (2 \cdot X_i^3 + R) / (3 \cdot X_i^2)$
42. Mostrar todos os números palíndromos de 5 algarismos. Números palíndromos (ou capicuas) são aqueles que lidos da esquerda para a direita ou vice-versa, resultam no mesmo número. Ex: 1331, 9999, 56765, etc

43. Calcular e escrever o valor de S dado por:

$$a. \quad S = \frac{1}{1} + \frac{3}{2} + \frac{5}{3} + \frac{7}{4} + \dots + \frac{99}{50}$$

44. Calcular e escrever a seguinte soma:

$$a. \quad S = \frac{37 \times 38}{1} + \frac{36 \times 37}{2} + \frac{35 \times 36}{3} + \dots + \frac{1 \times 2}{37}$$

45. Mostrar todos os números palíndromos de 5 algarismos. Números palíndromos (ou capicuas) são aqueles que lidos da esquerda para a direita ou vice-versa, resultam no mesmo número. Ex: 1331, 9999, 56765, etc

46. Calcular e escrever o valor de S onde:

$$a. \quad S = \frac{1}{1} - \frac{2}{4} + \frac{3}{9} - \frac{4}{16} + \frac{5}{25} - \frac{6}{36} \dots - \frac{10}{100}$$

47. Calcular e escrever a soma dos 50 primeiros termos seguinte série:

$$\frac{1000}{1} - \frac{997}{2} + \frac{994}{3} - \frac{991}{4} + \dots$$

48. Calcular e escrever a soma dos 30 primeiros termos da série:

$$\frac{480}{10} - \frac{475}{11} + \frac{470}{12} - \frac{465}{13} + \dots$$

49. 47) Calcular e escrever o valor do número  $\eta$ , com precisão de 0,0001, usando a série:

$$\eta = 4 - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} - \frac{4}{11} + \dots$$

50. Para obter a precisão pedida, somar apenas os termos com valor absoluto maior ou igual a 0,0001.

51. Calcular e escrever o valor de S no seguinte somatório:

$$a. \quad S = \frac{1}{225} - \frac{2}{196} + \frac{4}{169} - \frac{8}{144} + \dots + \frac{16384}{1}$$

52. Ler os dados de uma pesquisa a respeito de audiência de canais de televisão, sendo que cada entrevistado indicou o canal de TV que estava assistindo ao ser entrevistado, o seu sexo e a sua idade. Calcular a porcentagem de audiência de cada canal de televisão. Os canais pesquisados são: CH 2, CH 5, CH 7, e CH 9. Deve ser calculada também a porcentagem sobre outros canais. Determinar ainda quais são as quantidades e as porcentagens de mulheres e de homens por cada preferência. Determinar também qual é a quantidade e a porcentagem de crianças na idade de 5 a 8 anos por cada preferência. Finalmente, informar que canal tem maior audiência de crianças.

53. Solicitar que o operador digite letras vogais minúsculas. Caso seja digitado um caracter diferente, informar o erro e continuar. O programa deve parar quando o operador quiser. Ao final, informar quantas letras vogais foram digitadas e quantos erros foram cometidos.

54. Há três candidatos a uma vaga no senado. A contagem de votos será feita por computador. Fazer um programa para ler um conjunto de dados contendo, cada um, o voto de um eleitor. O último dado, que não deverá ser considerado nos cálculos, deverá ter valor  $-1$  (flag =  $-1$ ). Cada voto só poderá ter um dos seguintes valores:

- $-1$  - flag
- $0$  - voto em branco
- $1$  - voto para o candidato nº 1

- 
- 2 - voto para o candidato nº 2
  - 3 - voto para o candidato nº 3
  - 4 - voto anulado pelo eleitor
  - Calcular e mostrar uma tabela contendo os votos e a porcentagem dos votos válidos (voto anulado não é válido) de cada candidato.

55. Escreva programas usando a biblioteca <ctype.h> para manipulação de caracteres:

56. Ler um caractere e informar se ele é ou não alfabético (função isalpha).

57. Ler um caractere e informar se ele é ou não alfabético minúsculo (função islower). Se for, mostrar o correspondente maiúsculo (função toupper).

58. Ler um caractere e informar se ele é ou não alfabético maiúsculo (função isupper). Se for, mostrar o correspondente minúsculo (função tolower).

59. Ler um caractere e informar se ele é ou não alfanumérico (função isalnum).

60. Ler um caractere e informar se ele é ou não um dígito decimal (função isdigit).

61. Ler um caractere e informar se ele é ou não um caractere ASCII padrão (função isascii).

62. Ler um caractere e informar se ele é ou não um caractere de controle (função isctrl).

63. Ler um caractere e informar se ele é ou não um caractere gráfico (função isgraph).

64. Ler um caractere e informar se ele é ou não um caractere que pode ser impresso (função isprint).

65. Ler um caractere e informar se ele é ou não um símbolo de pontuação (função ispunct).

66. Ler um caractere e informar se ele é ou não um espaço (função isspace).

67. Escrever programas para:

68. Ler ou gerar aleatoriamente 15 números reais e informar:

- qual é o maior número;
- a média aritmética deles;
- quantos números são maiores que a média.

69. Ler ou gerar aleatoriamente os elementos (números inteiros) de uma matriz  $3 \times 3$  e:

- Mostrar a matriz lida, em forma de matriz (linhas e colunas);

- 
- Mostrar a soma dos elementos da diagonal principal;
  - Mostrar o produto dos elementos situados abaixo da diagonal principal;

70. Gerar aleatoriamente 20 números inteiros, positivos e diferentes, guardando-os em um vetor de 21 posições. Ordenar o vetor pela ordem crescente de seus elementos e mostrar esse vetor. Em seguida solicitar que seja digitado um número inteiro e positivo qualquer e verificar se ele já existe no vetor e em que posição. Se ele não existir, inseri-lo no vetor na posição correta, mantendo a ordem crescente, e mostrar como ficou o novo vetor. OBS: não vale inserir o número no final e re-ordenar o vetor!!!!
71. Ler as 10 notas dos alunos de uma turma. Escrever a nota média da turma e informar quantos alunos tiveram nota maior ou igual à média.
72. Gerar aleatoriamente, e mostrar em seguida, um conjunto de 100 números inteiros quaisquer, pertencentes ao intervalo  $[-500, +500]$ , não permitindo repetições de números.
73. Usando um vetor para guardar os restos das divisões, ler um número inteiro na base 10 e escrever esse número na base 2.
74. Idem, ler um número inteiro  $N$  na base 10 e em seguida um número  $B$  sendo que  $B$  ( $1 < B < 10$ ). Mostrar o número  $N$  na base  $B$ .
75. Idem, ler um número inteiro  $N$  na base 10 e mostrar  $N$  na Base 16 (hexadecimanal).
76. Ler os elementos inteiros de uma matriz  $4 \times 5$ . e:
- mostrar a matriz na tela dividida em linhas e colunas;
  - mostrar os elementos da terceira linha que são ímpares;
77. ler um número  $N$  qualquer e informar quantos elementos da matriz são:
- maiores que  $N$
  - iguais a  $N$
  - menores que  $N$
78. Ler duas séries de 05 números diferentes cada: uma contendo só números pares e outra contendo só números ímpares. Suponha que ambas as séries já estejam em ordem crescente. O programa deverá mostrar na tela uma única série com os 10 números em ordem crescente.

Ex. série de pares:            4            8            10            22            24

---

|                   |   |   |    |    |    |    |    |    |    |    |
|-------------------|---|---|----|----|----|----|----|----|----|----|
| série de ímpares: | 3 | 5 | 11 | 17 | 21 |    |    |    |    |    |
| série resultante: | 3 | 4 | 5  | 8  | 10 | 11 | 17 | 21 | 22 | 24 |

79. Ler dez letras quaisquer, repetidas ou não, e escrevê-las em ordem alfabética.
80. Gerar aleatoriamente um conjunto de 100 números inteiros, todos no intervalo [20,59], e guardá-los em um vetor. Informar quantas vezes cada número do intervalo acima aparece no conjunto gerado.
81. Gerar aleatoriamente um conjunto de 200 números inteiros no intervalo [0, 1000] e informar quantos terminam em 0, quantos terminam em 1, em 2, e assim por diante até 9.
82. Gerar aleatoriamente um conjunto de 200 números inteiros no intervalo [0, 1000] e mostrar um outro vetor eliminando as repetições do vetor gerado, isto é, cada número do vetor original só aparecerá uma vez no vetor pedido.
83. Idem o programa acima mas mostrando também as frequências com que cada número aparece no vetor original.
84. Escreva um programa para criar um vetor com capacidade para 100 elementos reais. Em seguida, ler um número inteiro  $N$  (sendo  $0 < N \leq 100$ ) representando a quantidade de elementos que será digitada e ler os valores digitados a seguir. Sem usar outro vetor, inverter a ordem dos elementos no próprio vetor e mostrá-lo.
85. Sabendo que em uma escola as notas finais variam de 0 a 100 e que são sempre números inteiros, escrever um programa para ler as notas (FLAG: nota = 101) dos alunos de uma turma e:
- calcular a média aritmética das notas da turma;
  - informar, para cada aluno da turma, qual é o desvio de sua nota em relação à média;
  - considerando apenas os valores absolutos dos desvios, calcular a média dos desvios.
86. Ler os elementos de uma matriz 2 X 3. Calcular e mostrar a sua matriz transposta.
87. Ler os elementos inteiros de uma matriz A[3][4] e os de outra B[4][2]. Mostrar as duas matrizes e em seguida mostrar a matriz produto A X B.
88. Gerar aleatoriamente os 200 elementos inteiros de um vetor e informar quantas vezes um determinado número (lido do teclado) aparece no vetor.
89. Gerar aleatoriamente os 200 elementos inteiros de um vetor A. Criar um vetor B só com os elementos de A que forem múltiplos de 3. Informar quantos elementos existem em B e mostrá-lo.

- 
90. Gerar aleatoriamente dois vetores A e B, cada um com 20 elementos inteiros e de forma que não exista repetições nos dois. Ordenar os dois vetores. Criar um vetor C intercalando os elementos de A e B de forma a obter o vetor C já ordenado.
91. Para se trabalhar com números inteiros muito grandes pode-se usar vetores de inteiros. Escreva um programa somar e mostrar dois números inteiros de até 30 algarismos lidos teclado.
92. Ler, algarismo por algarismo, um número inteiro de até 15 algarismos e mostrar o seu dobro.
93. Escreva um programa para o cálculo do salário líquido de um funcionário. Os cálculos são os seguintes:
- Salário Bruto = Horas trabalhadas \* Salário hora
  - Desconto = 0% do salário bruto, se este fosse menor que R\$ 100,00
  - Desconto = 10% do salário bruto, se este encontra-se entre R\$ 100,00 e 200,00 inclusive
  - Desconto = 20% do salário bruto, se este fosse maior a R\$ 200,00
  - Os valores de horas trabalhadas e salário hora deverão ser lidos do teclado. Sabe-se que o salário líquido é igual ao salário bruto menos o desconto.

**94. Para cada um dos exercícios propostos a seguir, escreva também o programa que irá “chamar” a função solicitada, passando para ela todos os dados necessários à sua execução e que também irá mostrar os resultados retornados pela funções.**

95. Escreva uma função para calcular o fatorial de um número inteiro e positivo N (passado para ela), de protótipo:

long int fatorial ( int N);

96. Escreva uma função de protótipo: int MDC (int A, int B);

que calcula e retorna o MDC de dois números inteiros A e B, usando o método das subtrações sucessivas, definido por:

- A se  $A = B$ ;
- $MDC ( A, B ) = \{ \quad MDC ( A - B, B )$  se  $A > B$
- $MDC ( A, B - A )$  se  $B > A$

97. Escrever uma função de protótipo

**int strchr ( char s[], char ch )**

que procura em uma cadeia de caracteres 's', a primeira ocorrência do caracter 'ch' e retorna a posição em que ele foi encontrado ou retorna '-1' se não existir o caracter na cadeia.

---

98. Escrever uma função de protótipo

**void strinsere ( char s[], char ch, int pos )**

que insere, em uma cadeia de caracteres 's', o caracter 'ch' na posição 'pos'.

99. Escreva uma função de protótipo:

**int achaalg (long int A, int B);**

que recebe um número inteiro A, de até 8 algarismos, e um numero inteiro B, de apenas um algarismo, e retorna a posição em que o algarismo B aparece, pela primeira vez, em A. A função deve retornar '-1' se não encontrar o algarismo B em A.

100. Escreva um programa que irá ler do teclado um número inteiro A, de até 8 algarismos, e um número inteiro B, de apenas um algarismo. O programa deverá usar a função definida acima para retornar a posição em que o número B (ou algarismo B) aparece pela primeira vez no número A.

101. Escreva uma função de protótipo:

**int contarac (char A[5][4], char B);**

que recebe uma matriz de caracteres A e um caracter B e retorna quantas vezes o caracter B aparece na matriz A.

102. Escreva um programa que irá ler do teclado os elementos da matriz A [5] [4] e um caracter B. O programa deverá usar a função definida acima para retornar quantas vezes o caracter B aparece na matriz A.

103. Escreva uma função de nome *soma* ( ) que receba um número inteiro e positivo N como argumento e retorne a soma dos N primeiros números inteiros e positivos.

104. Escreva uma função chamada *potência* ( ) que aceite dois argumentos inteiros e positivos i e j. A função deverá retornar o valor de i elevado à potência j. Lembre-se que uma potência é um produto de fatores iguais.

105. Faça também a função *produto* ( ) que aceite dois argumentos inteiros e positivos i e j e retorne i multiplicado por j. Lembre-se que um produto é uma soma de parcelas iguais.

106. Sabe-se que a Série de Fibonacci é uma seqüência infinita de números em que:

- o primeiro termo é 1 (um);
- o segundo termo é 1 (um);
- a partir do terceiro, cada termo é igual à soma dos dois termos imediatamente anteriores.



- Assim, tem-se a Série de Fibonacci:  $1\ 1\ 2\ 3\ 5\ 8\ 13\ 21\ 34\ 55\ 89\ \dots$

Escreva uma função que recebe um número inteiro que indica a posição de um termo na Série de Fibonacci e retorne esse termo. Por exemplo: se a função receber o número 3, ela deve retornar 2 que é o terceiro termo da série; se ela receber o número 8, ela deve retornar 21, que é o oitavo termo da série.

107. Escreva uma função de protótipo: `int max ( int vet[ ], int tam);`

que recebe um vetor de inteiros e seu tamanho e retorna o maior elemento dele.

108. Escreva uma função de protótipo: `int min ( int vet[ ], int tam);`

que recebe um vetor de inteiros e seu tamanho e retorna o menor elemento dele;

O que poderia ser feito para que as duas funções dos exercícios anteriores fossem incorporadas em uma só? Mostre como ficaria esta função.

109. Sabe-se que uma progressão aritmética é uma seqüência infinita de números em que a diferença entre um termo e o seguinte a ele é constante e recebe o nome de razão.

Por exemplo:  $2\ 5\ 8\ 11\ 14\ 17\ 20\ 23$

é uma progressão aritmética cujo primeiro termo é 2 e a razão 3.

Escreva uma função que recebe três valores inteiros: - 'n' um número que indica a posição de um termo; 'a' o termo inicial da progressão e 'r' a razão da progressão e retorna o n-ésimo termo da progressão. Por exemplo: se a função receber os valores 3, 2, 3, nesta ordem, ela deve retornar 8 que é o terceiro termo da progressão mostrada acima; se ela receber os números 8, 2, 3 ela deve retornar 23, que é o oitavo termo da progressão.

110. Escreva uma função com o protótipo: `int IDIV ( int Dividendo, int Divisor ),`

que retorna o quociente inteiro de dois números, sabendo que uma divisão inteira é uma seqüência de subtrações.

Isto é, na divisão inteira verifica-se quantas vezes o DIVISOR "cabe" no DIVIDENDO.

111. Escreva uma função com o protótipo: `int IRESTO ( int Dividendo, int Divisor ),`

que retorna o resto inteiro de dois números, usando o mesmo conceito de divisão mostrado acima.

112. Escreva uma função para comparar dois vetores de inteiros do mesmo tamanho, passados como parâmetros para ela. A função deve retornar "1" se os dois vetores forem iguais ou "0" se não forem. Para um vetor ser igual a outro, de mesmo tamanho, TODOS os elementos de um devem ser respectivamente iguais aos elementos do outro.

---

113. Escreva uma função para calcular com 20 parcelas:

$$\cos(x) = 1 - x^2/2! + x^4/4! - x^6/6! + x^8/8! - \dots$$

onde “x” é uma medida em radianos

114. Escreva uma função de protótipo: `void sub_string ( char s1[ ], int i, int j, char s2[ ] )`

que devolve em “s2” uma “sub-string” de “s1” com “j” caracteres a partir do caracter começando em “s[i]”.

115. Escreva uma função que recebe duas “strings” como argumentos e troca o conteúdo da primeira “string” com o conteúdo da segunda.

116. Escreva uma função que recebe uma string de caracteres “str” e um outro caracter “ch” e retira de “str” todos os caracteres iguais a “ch”.

Exemplo: String recebida: “arara”

Caracter recebido: “r”                      String resultante:                      “aaa”

117. Escreva uma função chamada COME\_LETRA que recebe uma string e um caractere e retira da string todas as ocorrências do caractere na string.

Ex. COME\_LETRA (“abacate”, ‘a’) retorna → bcte

118. Escreva uma função que recebe uma seqüência (vetor) de números inteiros e inverta a ordem de seus elementos.

119. Escrever uma função para retornar a soma dos quadrados dos números inteiros e positivos de 1 até N . O valor de N deverá ser passado para ela.

120. Usando uma função que recebe um número inteiro qualquer e retorna o seu fatorial (vide exercício 1), escrever um programa para ler do teclado um número inteiro N (sendo  $N \leq 10$ ) e calcular e escrever a soma S definida abaixo:

$$S = 1 + (3 / 2!) + (5 / 3!) + (7 / 4!) + \dots + ((2*N - 1) / N!)$$

121. Escreva uma função de protótipo:

`float distancia ( float ax, float ay, float bx, float by )`

onde “ax”, “ay”, “bx” e “by” são as coordenadas cartesianas de dois pontos “a” e “b”. A função acima deve retornar a distância entre os pontos “a” e “b”.

122. Agora, escreva um programa para ler do teclado as coordenadas dos vértices de um triângulo e, usando a função acima, o programa deve calcular o perímetro e a área do triângulo.

123. Escrever um programa para ler as medidas de três segmentos de reta. Usando uma função que recebe as três medidas e retorna: (0) se os segmentos não formarem um triângulo; (1) se os três segmentos formarem um triângulo equilátero, (2) se os três segmentos formarem um triângulo isósceles e (3) se os três segmentos formarem um triângulo escaleno; o programa deve informar se os três segmentos formam ou não um triângulo e de que tipo ele é.

124. Escreva um programa que irá ler três números inteiros: N, A e R, respectivamente: o número de termos, o primeiro termo e a razão de uma progressão aritmética e irá escrever os N primeiros termos da progressão aritmética correspondente, utilizando a função pedida no exercício 13.. Ao final, o programa deverá fornecer a soma dos termos da seqüência mostrada.

125. Escreva um programa (e as funções correspondentes) que:

- Lê do teclado um número inteiro N, fornecido pelo usuário, podendo o valor de N ser qualquer número entre 10 e 200.
- Cria um vetor de N elementos inteiros sendo que esses elementos são gerados aleatoriamente dentro do intervalo [1,100].

126. Obtém, usando uma função (cujo protótipo é fornecido abaixo), a média dos valores gerados no vetor acima, guardando esta média em uma variável de nome “media”;

127. Informa, usando para isso outra função (cujo protótipo também é fornecido abaixo), quantos elementos do vetor são maiores que a média deles.

```
float MEDIA_VET (int Vet [], int Tam);
```

```
int ACIMA_MED (int Vet [], int Tam, float media);
```

128. Escreva um programa (e as funções correspondentes) que:

- Cria um vetor de 100 elementos inteiros sendo que esses elementos são gerados aleatoriamente dentro do intervalo [1,1000] (use as funções randomize() e random());
- Informa, usando para isso duas funções (cujos protótipos são fornecidos abaixo), qual é o maior e qual é o menor elemento do vetor criado acima.

```
int MAX (int Vet [], int Tam);
```

```
int MIN (int Vet [], int Tam);
```

---

**Referências Bibliográficas**

1. Mizrahi, Victorine Viviane. *Treinamento em Linguagem C++*. Módulo 1. São Paulo. Makron Books, 1994.
2. Material do professor Eduardo Monteiro de Andrade da PUC/Minas
3. Material da professora Cíntia da PUC/Minas