Algoritmos e Estrutura de Dados

Lista de exercícios

1. Escreva um programa que lê uma matriz e depois verifica se esta é uma matriz triangular inferior, como a do exemplo abaixo:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 2 & 3 \end{bmatrix}$$

2. Escreva um programa que leia um valor n da entrada padrão e preencha uma matriz $n \times n$ de tal forma que esta se torne a matriz identidade I_n. Ao final, imprima a matriz na tela. Considere $0 < n \leq 100$.

3. Escreva um programa que leia da entrada padrão uma matriz de tamanho $n \times m$ (n e m fornecidos pelo usuário) e escreva na saída padrão o valor máximo e o valor mínimo existentes naquela matriz. Considere $0 < n \leq 100$ e $0 < m \leq 100$.

4. Considere o seguinte esqueleto de um código fonte C++:

```
using namespace std;

typedef float prova;
typedef float media;

struct Provas{
        prova p[3];
        media M;
};

struct Aluno{
        string matricula;
        int frequencia;
        Provas Ps;
};


int main() {
        int n,i,j;
        float desvios_Ps[3]={0.0, 0.0, 0.0}, medias_Ps[3]={0.0, 0.0, 0.0};
        Aluno alunos[MAX_ALUNOS];

        cin >> n;
```

```
/* COMPLETE AQUI */

}
```

Complete o esqueleto acima. O programa deve ler um número inteiro n que indicará a quantidade de alunos na turma. Após isso, deve ler as notas das provas de cada aluno, calcular e mostrar na tela as notas e a média de cada aluno, a média geral de cada prova de todos os alunos e o desvio padrão de cada prova de todos os alunos. Note que não é necessária a declaração de mais nenhuma variável. Exemplo:

```
ENTRADA:
3
030034
15
4 5 6
123456
12
0 5 10
987654
10
6 9 3

SAIDA:
Matricula: 030034
Prova 0: 4.000000
Prova 1: 5.000000
Prova 2: 6.000000
Média: 5.000000
----------------
Matricula: 123456
Prova 0: 0.000000
Prova 1: 5.000000
Prova 2: 10.000000
Média: 5.000000
----------------
Matricula: 987654
Prova 0: 6.000000
Prova 1: 9.000000
Prova 2: 3.000000
Média: 6.000000
----------------
Média geral P0: 3.333333
Desvio padrão P0: 2.494438
Média geral P1: 6.333333
Desvio padrão P1: 1.885618
Média geral P2: 6.333333
Desvio padrão P2: 2.867442
```

5. Cite as vantagens da utilização de funções em um programa.

6. Escreva uma função que soma dois números inteiros com passagem de parâmetro por valor. Escreva a mesma função utilizando as duas formas de passagem de parâmetro por referência do

C++. Para cada uma das funções escreva como fica a chamada da função no programa principal.

7. O que caracteriza uma variável local? E uma global? Descreva os escopos desses dois tipos de variáveis.

8. Qual a saída do programa abaixo para a entrada 3 4 5? Explique passo a passo a execução do programa com essas entradas, mostrando os valores das variáveis **em cada linha**, desde o início do programa principal. Explique as declarações e depois siga a sequência de execução do programa para explicar passo a passo, utilizando os números das linhas para numerar sua explicação. Portanto, inicie explicando a linha 1, depois linha 2, depois linha 7, linha 8, e assim por diante. Todas as linhas numeradas devem ser explicadas, e sua explicação deve ter a mesma sequência que a execução do programa.

```
  #include<iostream>
1. int z=0;
2. void troca(int *x, int *y) {
3.       int z;
4.       z=*x;
5.       *x=*y;
6.       *y=z;
   }

  int main() {
7.       int a,b;
8.       std::cin >> a >> b >> z;
9.       troca(&a,&b);
10.      std::cout << a << b << z << std::endl;
  }
```

9. Qual o erro no programa abaixo? Porque esse erro impede o funcionamento do programa? Descreva uma correção alterando apenas declarações de variáveis e funções. Descreva outra correção alterando apenas código que não é declaração de variável nem declaração de função.

```
#include<iostream>
#define TAM 10
//atencao: note que declarar int *vetor equivale declarar int vetor[]
void copia_vetor(int *vetor, int *vetor2)
{
   int vetor_aux[TAM], i;
   for(i=0; i < TAM; i++)
      vetor_aux[i] = vetor[i];

   vetor2=vetor_aux;
}

void imprime_vetor(int *vetor)
{
   int i;
   for(i=0; i < TAM; i++)
      std::cout << vetor[i] << " ";
   std::cout << "\n";
```

```
}

int main()
{
    int vet[TAM], vet2[TAM],i;

    for(i=0; i<TAM; i++)
       std::cin >> vet[i];

    copia_vetor(vet, vet2);
    imprime_vetor(vet);
    std::cout << "Cópia:";
    imprime_vetor(vet2);
}
```

10. Considere o trecho de programa abaixo. Depois de executado o trecho inteiro, quais os valores associados aos itens de (a) a (g)? Suponha que os endereços das variáveis u e v são 1000 e 1004, respectivamente.

```
int u,v;
int *pv,*pu;
v = 45;
pv = &v;
*pv = v + 1;
u = *pv + 1;
pu = &u;
```

(a)&v
(b)pv
(c)*pv
(d)u
(e)&u
(f)pu
(g)*pu

11. Escreva uma **função** que recebe como parâmetros um operando (inteiro), um operador aritmético (char) e outro operando (inteiro) e retorna o cálculo da operação indicada. As operações possíveis são soma(+), subtração(−), multiplicação(*), divisão(/) e resto da divisão inteira(%). Escreva também a função `main()` com a chamada à função criada para poder testá-la.

12. Faça uma **função** em linguagem C++ que recebe um inteiro n, calcula o valor da soma abaixo e retorna a soma calculada. Escreva também a função `main()` com a chamada à função criada.

$$1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{16} + \frac{1}{32} + ... + \frac{1}{2^{n-1}}$$

13. Escreva uma **função** em C++ que recebe como parâmetro uma string e um caractere, e retorna como resultado o número de ocorrências desse caractere na cadeia passada como parâmetro. O cabeçalho da função deve ser `int conta (std::string s, char c);`

14. Escreva uma **função** em C++ que recebe como parâmetro uma string, e retorna como resultado o número de vogais na cadeia passada como parâmetro. O cabeçalho da função deve ser `int contavogais (std::string s);`

15. Escreva uma **função** em C++ que recebe como parâmetro uma string, e retorna como resultado o número de consoantes na cadeia passada como parâmetro. O cabeçalho da função deve ser `int contaconsoantes (std::string s);`

16. Considere o trecho de programa abaixo. Depois de executado o trecho inteiro, quais os valores associados aos itens de (a) a (i)? Suponha que os endereços das variáveis a, b e c são 1000, 1004 e 1008, respectivamente.

```
float a,b;
float c, *pa, *pb;
a=0.001;
b=0.003;
pa = &a;;
*pa = 2 * a;
pb = &b;
c = 3 * (*pa + *pb);
```

(a)&a
(b)&b
(c)&c
(d)pa
(e)*pa
(f)&(*pa)
(g)pb
(h)&(*pb)
(i)c

17. Dado p um ponteiro qualquer, explique a diferença entre `p++, (*p)++` e `*(p++)`.

18. Escreva um programa para armazenar um certo número de inteiros um vetor. O programa pergunta inicialmente o número de inteiros a serem guardados, e então aloca dinamicamente um vetor de inteiros capaz de armazenar esse número de inteiros. Então o programa deve ler os inteiros e imprimi-los na tela. Não esqueça de liberar a memória usada antes de terminar o programa.

19. Escreva a função `int *multiplica_vetores(int A[], int B[])`, que recebe como parâmetro dois vetores A e B que contêm 10 números inteiros cada, aloca dinamicamente um vetor C e, para cada posição i, calcula $A[i] * B[i]$ e guarda em $C[i]$. A função deve retornar um apontador para o vetor C.

20. Descreva com suas próprias palavras o que a função `misterio` abaixo faz. A sua descrição deve conter: (a) o que são (o que representam e qual o conteúdo) os parâmetros passados para a função; (b) o significado dos parâmetros passados à função `open`; (c) uma descrição da lógica utilizada no `while`.

```
int misterio(std::string nome1, std::string nome2) {
  fstream fr, fw;
  std::string b;

  fr.open(nome1, ios::in);
  fw.open(nome2, ios::out);
  while (getline(fr, b))
    fw << b << std::endl;

  fr.close();
  fw.close();
  return 0;
}
```

21. Explique, com suas palavras, o que é uma classe e o que é um objeto, e a diferença entre os dois.

22. Explique, com suas palavras, o que são atributos e métodos de um objeto.

23. Escreva uma classe retangulo, que representa a forma geométrica de um retângulo.

(a) Dois construtores, o default (sem argumentos) e com os argumentos lado e altura fornecidos

(b) Deve conter os atributos privados lado e altura, deve conter getters/setters para eles e método que retorna a área do retângulo

24. Escreva um programa que cria objetos da classe retangulo e testa seus valores, calcula a área do objeto associado, mostrando o resultado na saída padrão.

25. Escreva uma outra classe chamada paralelepipedo, que é subclasse de retangulo, e que representa a forma geométrica de um paralelepípedo. Deve conter o atributo profundidade, getter/setter para a profundidade e método que retorna o volume do paralelepípedo.

26. Atualize o programa da questão 24 para criar objetos paralelepipedo e testá-los.

27. Escreva uma classe MinhaString para representar cadeias de caracteres. Esta classe deve receber em seu construtor um **vetor de caracteres (char)** e não deve fornecer métodos para que o valor inicializado neste vetor se altere. Desenvolva (pelo menos) três métodos públicos para esta classe:

    (a) `tamanho()`: retorna o tamanho do array de caracteres que compõe o objeto.

    (b) `compara()`: compara o objeto MinhaString com outro objeto MinhaString e retorna verdadeiro se o valor da cadeia de caracteres de cada objeto foi idêntico, e falso caso contrário.

    (c) `printMe()`: imprime a cadeia de caracteres do objeto na tela.