

Lista de exercícios – Revisão de programação C

Professor: Adelardo Adelino Dantas de Medeiros

- 1) Fazer um programa no qual o usuário vai entrando sucessivamente com valores positivos ou nulos. Quando o usuário entrar com um valor negativo, o programa para de pedir valores e calcula a média dos valores já fornecidos (excluindo o último número negativo)
- 2) Fazer um programa que recebe um símbolo de operação do usuário (+, -, / ou *) e dois números reais. O programa deve retornar o resultado da operação recebida sobre estes dois números. Este funcionamento deve ser repetido até que o símbolo seja algo que não corresponda a nenhuma das quatro operações reconhecidas.
- 3) Fazer um programa que sorteie um número de 0 a 100 e que permita que o usuário (sem conhecer o número sorteado) tente acertar. Caso não acerte, o programa deve imprimir uma mensagem informando se o número sorteado é maior ou menor que a tentativa feita. Ao acertar o número, o programa deve imprimir a quantidade de tentativas feitas.
- 4) Qual o resultado da execução do seguinte programa, supondo que o usuário digite os valores 3 e 4? De uma maneira geral, o que faz este programa?

```
#include <iostream>
using namespace std;
void main()
{
    int x, y, p, a;
    cout << "Informe dois números:";
    cin >> x;
    cin >> y;
    p = 0;
    a = y;
    while(a > 0){
        p += x;
        a--;
    }
    cout << "\nResultado: " << p << endl;
}
```

- 5) Fazer uma função que recebe um dia, um mês e um ano como parâmetros e retorna o número de dias desde o início daquele ano. Para o dia 1º de janeiro de qualquer ano, a função deve retornar 1. Para o dia 31 de dezembro, ela deve retornar 365 ou 366, dependendo de o ano ser ou não bissexto. Um ano é bissexto quando é múltiplo de 400 ou também quando é múltiplo de 4 e não múltiplo de 100.
- 6) Faça um programa que, dado um valor em reais (sem centavos), mostre a menor combinação existente de notas (R\$ 100, 50, 20, 10, 5 e 2) para esse valor.
- 7) Fazer uma função que receba 3 números como parâmetros, A, B e C, e os ordene de tal forma que, ao final da função, A contenha o menor número e C o maior. Em seguida, fazer um programa que receba 3 números do usuário, chame a função e mostre os números ordenados.
- 8) Desenvolver a função INVERTE que recebe um número (`unsigned int`) como parâmetro e retorna este número invertido. Ex: 431 retorna 134.

- 9) Crie uma função que receba como parâmetro um vetor de `float`, definido de tal forma que o último número seja zero. Esta função deve retornar a diferença entre a média dos números contidos neste vetor e o menor dos números, não contando com o zero final.
Exemplo: Vetor: (1, 7, 4, 0); Média: 4; Menor: 1; Retorno = média - menor = 3;
- 10) Fazer um programa que receba do usuário uma quantidade N de números a ser digitada. Em seguida, o programa deve solicitar N números do usuário. No final, o programa exibe o máximo, o mínimo e a média dos valores digitados e quantos valores estão acima da média.
- 11) Faça um programa para receber do usuário a dimensão e os elementos de um vetor e efetuar a sua ordenação utilizando o método da bolha. Neste método de ordenação, percorre-se o vetor comparando cada elemento com o seguinte: se eles estiverem na ordem errada, os elementos são permutados. O vetor é percorrido tantas vezes quantas necessárias até que não seja necessário mudar nenhum elemento.
- 12) Faça um programa que, dados dois vetores bidimensionais (matrizes A e B) com dimensões de 5x5 elementos, imprima a soma destas duas matrizes, a soma das diagonais de cada matriz e a multiplicação das duas matrizes.
- 13) Seja `a[]` um vetor qualquer, independente de tipo e tamanho, e `pa` um ponteiro para o mesmo tipo de `a[]`. Verifique se as afirmações a seguir são verdadeiras ou falsas:
- Após a atribuição `pa=a[0]`, `pa` e `a` possuem valores idênticos
 - A atribuição `pa=&a[0]` pode ser escrita como `pa=a`
 - Uma referencia a `a[i]` pode ser escrita como `&(a+i)`
 - `&a[i]` e `a+i` são idênticos
 - `a+i` é o endereço do i-ésimo elemento do vetor
 - `pa[i]` é idêntico a `* (pa+i)`
 - `pa=a` é uma operação válida
 - `pa++` é uma operação válida
 - `a=pa` é uma operação válida
 - `a++` é uma operação válida
- 14) Qual o erro no trecho de programa a seguir?

```
/* esse trecho de programa está errado */  
int x, *p;  
x = 10;  
*p = x;  
cout << *p;
```

- 15) Qual o resultado impresso após a execução do trecho de programa a seguir?

```
int i, j, *p;  
i = 4;  
j = 7;  
p = &j;  
*p = 9;  
cout << i << " " << j << " " << *p;
```