

## Lista de Exercícios 7

### Informações sobre cópias

As questões são individuais. Em caso de cópias de trabalho a pontuação será zero para os autores originais e copiadores. Não serão aceitas justificativas como: “Fizemos o trabalho juntos, por isso estão idênticos”.

### Dicas

1) Exemplo de uso da Função *rand* para geração de número aleatórios entre 0 e 99.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4
5  int main(void){
6      // comando para gerar numeros aleatorios diferentes
7      srand(time(0));
8
9      printf("\nDez numeros aleatorios entre 0 e 99: ");
10     for(int i=0; i<10; i++)
11         printf(" %i", rand() % 100);
12     return 0;
13 }
```

### **Parte 1 – Exercícios para serem entregues**

Resolva os exercícios a seguir e entregue pelo CANVAS. Cada exercício deve conter um arquivo no **formato .C**. O uso apropriado de funções e procedimentos serão considerados na avaliação.

1. Escreva um programa para preencher um vetor com 20 vinte valores inteiros (os valores podem ser lidos do teclado ou gerados automaticamente). Em seguida, o sistema deve solicitar ao usuário um valor, que deve ser pesquisado no vetor. Imprima as posições do vetor que armazena o valor informado.

2. Escreva um programa que preencha um vetor de tamanho 100 com os 100 primeiros números naturais que não são múltiplos de 6 e que não terminam com 6.  
Atenção: todas as 100 posições do vetor devem ser preenchidas.

3. Escreva um algoritmo que:

- (a) Crie um arranjo de 5 elementos inteiros e o preencha de números
- (b) Procure a posição do menor elemento deste arranjo
- (c) Troque o menor elemento com elemento da primeira posição
- (d) Imprima os elementos do arranjo

*Cada um dos itens acima devem ser implementados em funções ou procedimentos separados que recebem o vetor por parâmetro.*

4. Escreva um programa que receba do usuário dois vetores, A e B, com 10 números inteiros cada. Crie um novo vetor, C, calculando  $C = A - B$ . Mostre na tela os dados do vetor C.

5. Implemente um procedimento *preencheValores* que preencha um vetor X de 10 elementos. Na sequência, faça um procedimento *copiaNegativos* que receba um vetor preenchido, teste e copie todos os valores negativos deste vetor para um novo vetor, sem deixar elementos vazios entre os valores copiados. O vetor contendo números negativos deve conter até 10 elementos - após o último número negativo (caso não existam 10 número negativos) o vetor deve conter o número 0. Faça um programa (main) que acione os procedimentos e imprima o vetor de números negativos, sem imprimir o valor zero.

```
void preencheValores(int vetor[], int length);  
void copiaNegativos(int vetor[], int length, int vetorNeg[]);
```

6. Escreva um programa que preencha e imprima na tela um vetor de tamanho 50, cujos valores são dados pela seguinte fórmula:

$$pos = (i + (3 \times i)) \% (i + 1)$$

onde i corresponde à posição do elemento no vetor.

7. Escreva um programa que leia da entrada padrão 5 números reais, que devem ser armazenados em um arranjo. Em seguida, gere outro arranjo, cujos valores correspondem ao dobro dos respectivos elementos do primeiro arranjo.

8. Escreva um vetor de inteiros A com 10 elementos. Em seguida, leia um vetor de inteiros B, também com 10 elementos. Durante a leitura do vetor B, se o elemento na posição i for igual ao elemento na mesma posição do vetor A, rejeitar o valor e solicitar a entrada de um novo valor. A rejeição deve ocorrer até que o valor seja digitado um valor válido. Após a leitura dos vetores A e B, criar um algoritmo que calcule e imprima a diferença entre cada um dos elementos.

9. Escreva um programa que armazene as idades dos alunos que estão presentes em uma aula da disciplina de Algoritmos e Estrutura de Dados I. Considere que o vetor possa conter até 60

registros. Sabe-se que, em uma dada aula, alguns alunos podem ter faltado. Com isso, leia elementos até que seja digitado um valor 0 ou enquanto a quantidade de alunos for inferior à capacidade do vetor. Imprima:

- Idade de todos os alunos presentes na aula (não imprimir idades não cadastradas).
- Idade de todos os alunos com idade superior à média.