

09/09/2024

Eletrônica 4.0

Módulo 3 – Lista de Exercício 5

INDT - Instituto de Desenvolvimento Tecnológico

Elaborador por: [Diego Machado](#)



7.1. Escreva um programa que leia as temperaturas de uma semana inteira em um array `temperature[]` e imprima a temperatura média dessa semana.

Digite a temperatura para o dia 0: 5

Digite a temperatura para o dia 1: 5

Digite a temperatura para o dia 2: 5

Digite a temperatura para o dia 3: 5

Digite a temperatura para o dia 4: 6

Digite a temperatura para o dia 5: 6

Digite a temperatura para o dia 6: 6

A temperatura média para esta semana é 5.43

7.2. Altere o exercício anterior de forma que todos os dias com uma temperatura superior a 10°C sejam impressos.

Digite a temperatura para o dia 0: 1

Digite a temperatura para o dia 1: 8

Digite a temperatura para o dia 2: 10

Digite a temperatura para o dia 3: 12

Digite a temperatura para o dia 4: 15

Digite a temperatura para o dia 5: 14

Digite a temperatura para o dia 6: 9

Todos os dias com temperatura > 10°C: dia 03

dia 04

dia 05



7.3. Repita o exercício 7.1 usando 2 funções. Crie uma função para ler as temperaturas e uma função separada para calcular a temperatura média. A impressão da média deve ser feita no programa principal.

7.4. Repita o exercício 7.2, mas desta vez, use 3 funções separadas para ler as temperaturas (ReadTemp), determinar quais dias estão mais quentes que 10°C (Calculate) e imprimir os dias encontrados (PrintDays). Extra: peça ao usuário para inserir um limite de temperatura.

7.5. Escreva um programa com a função principal:

```
#include <stdio.h>
#define SIZE 12
#define COLUMNS 3
void ReadArray(int [], int);
void PrintMatrix(int [], int, int);
int main(void)
{   int a[SIZE];
    ReadArray(a, SIZE);
    PrintMatrix(a, SIZE, COLUMNS);
    return 0;
}
```



- A função ReadArray lê 12 (SIZE) números e os armazena no array unidimensional a.
- A função PrintMatrix imprime os 12 números em 4 linhas de 3 (COLUMNS) números cada. Adaptando apenas SIZE e COLUMNS, o programa deve também ser capaz de imprimir, por exemplo, 20 números em uma matriz 5 x 4.

7.6. Altere o exercício 7.5 para que, após cada linha, a soma de todos os elementos daquela linha seja impressa.

7.7. Altere o exercício 7.6 para que, sob cada coluna, a soma de todos os elementos daquela coluna seja impressa.



7.8. Escreva um programa com uma função principal e 3 funções adicionais.

- Na função principal, um array de 100 inteiros é declarado e as 3 funções adicionais são chamadas.
- A primeira função lê 1 inteiro no intervalo [0, 100]. Este inteiro representa o número efetivo de elementos que o array conterá. Esse inteiro precisa ser retornado ao programa principal.
- A segunda função lê o número correto de inteiros e os armazena no array.
- A terceira função imprime os inteiros lidos anteriormente. A função principal é parecida com:

```
int main(void)
{
    int row[MAX];
    int size;
    size = ReadSize(0, MAX);
    ReadArray(row, size);
    PrintArray(row, size);
    return 0;
}
```



7.9. O mesmo que o exercício 7.8, mas sem pedir ao usuário para inserir o número efetivo de inteiros. O usuário insere os inteiros e termina com 999 para indicar o fim do processo de leitura. 999 NÃO pode ser armazenado no array! A função principal é parecida com:

```
int main(void)
{
    int row[MAX];
    int size;
    size = ReadArray(row);
    PrintArray(row, size);
    return 0;
}
```

7.10. Adicione uma função extra ao exercício 7.9 que calcula o valor médio de todos os inteiros inseridos e retorna esse valor médio para a função principal. A média deve ser impressa na função principal. O cabeçalho dessa função extra pode ser assim:

```
double CalcMean(int row[], int size)
```

7.11. Repita o exercício 7.10, mas desta vez, escreva uma função que procura o valor máximo de todos os inteiros inseridos e retorna esse valor máximo para a função principal.



7.12. Escreva um programa com funções que mescla 2 listas ordenadas (ordenadas do menor para o maior) em 1 lista ordenada.

Digite a lista ordenada: 5 9 12 54 78

Digite a lista ordenada: 2 3 34 45 102

A lista mesclada é: 2 3 5 9 12 34 45 54 78 102

7.13. Adicione uma função extra ao exercício 7.9 ou 7.10 para que, antes da impressão, os maiores e menores inteiros sejam trocados de lugar. A troca deve acontecer dentro do array, não use um segundo array. O uso de uma variável auxiliar é permitido.

7.14. Escreva um programa com funções:

- A primeira função lê 20 números e os armazena em um array.
- A segunda função garante que nenhum dos números no array ocorra mais de uma vez. Para isso, a segunda, terceira, ... ocorrência de um número é removida do array e todos os outros elementos do array são deslocados para a esquerda.

Digite uma lista de 20 números:

2 5 3 4 6 5 3 4 6 7 2 4 4 5 3 2 1 6 1 7

Os números armazenados no array limpo são:

2 5 3 4 6 7 1



7.15. Escreva um programa que gera 6 números aleatórios diferentes no intervalo [1, 42]. Use um array para armazenar os números e garantir que todos os 6 números sejam diferentes.

7.16. Escreva um programa com funções. Uma primeira função lê 2 listas de no máximo 10 inteiros positivos. O processo de leitura para quando um 0 é inserido ou quando 10 inteiros são inseridos. (não armazene o número 0!) Uma segunda função verifica se ambas as listas contêm números iguais.

Digite uma lista de no máximo 10 inteiros (pare com 0):

4 4 12 9 5 0

Digite uma lista de no máximo 10 inteiros (pare com 0):

9 9 4 12 4 5 9 12 4 4

As listas:

4 4 12 9 5 0

e

9 9 4 12 4 5 9 12 4 4

contêm números iguais.



7.17. Escreva um programa que simula o lançamento de 2 dados. Lançar 2 dados sempre resulta em um valor entre 2 e 12. Quantas vezes cada valor possível ocorrerá se os dados forem lançados 400 vezes?

- Declare um array na função principal para manter as ocorrências de cada valor possível.
- Uma primeira função simula 400 lançamentos com 2 dados e calcula os valores. (dica: `rand()%6+1` resulta em um número aleatório de 1 a 6, simulando o lançamento de 1 dado).
- Uma segunda função imprime as ocorrências.
- Extra: imprima um gráfico de barras horizontal com as ocorrências. Use o símbolo # para desenhar as barras. Use uma função `DrawLine` para realizar esta parte do programa.

Espero que essas traduções sejam úteis! Se precisar de mais alguma coisa, estou à disposição.



 @indt.instituto

 [linkedin.com/company/indt-instituto](https://www.linkedin.com/company/indt-instituto)

www.indt.org.br