

Atividade Pontuada: Implementar Listas UFBA

1. Escreva um programa que receba uma lista de 10 inteiros via teclado e imprima toda a lista em uma mesma linha.

```
#include <stdio.h>

int main() {
    int numeros[10];

    printf("Digite 10 números inteiros:\n");
    for(int i = 0; i < 10; i++) {
        scanf("%d", &numeros[i]);
    }

    printf("Os números digitados foram:\n");
    for(int i = 0; i < 10; i++) {
        printf("%d ", numeros[i]);
    }

    return 0;
}
```

2. Escreva um programa que receba uma lista de 10 inteiros via teclado, em seguida o programa deve solicitar um número e informar se o número também está na lista ou não.

```
#include <stdio.h>

int main() {
    int numeros[10];
    int numero, encontrado = 0;

    printf("Digite 10 números inteiros:\n");
    for(int i = 0; i < 10; i++) {
        scanf("%d", &numeros[i]);
    }
    printf("Digite um número para verificar se está na lista:\n");
    scanf("%d", &numero);

    for(int i = 0; i < 10; i++) {
        if(numeros[i] == numero) {
            encontrado = 1;
            break;
        }
    }
    if(encontrado) {
        printf("O número %d está na lista.\n", numero);
    } else {
        printf("O número %d não está na lista.\n", numero);
    }

    return 0;
}
```

3 Escreva um programa que leia 10 números de reais e informe:

- a) a média dos elementos;
- b) o maior e menor elemento;
- c) a quantidade de elementos positivos e a quantidade de elementos negativos.

```
#include <stdio.h>

int main() {
    float numeros[10], soma = 0, maior, menor, media;
    int positivos = 0, negativos = 0;

    printf("Digite 10 números reais:\n");
    for(int i = 0; i < 10; i++) {
        scanf("%f", &numeros[i]);
        soma += numeros[i];
        if(i == 0) {
            maior = menor = numeros[i];
        } else {
            if(numeros[i] > maior) maior = numeros[i];
            if(numeros[i] < menor) menor = numeros[i];
        }
        if(numeros[i] > 0) positivos++;
        else if(numeros[i] < 0) negativos++;
    }

    media = soma / 10;

    printf("A média dos elementos é: %.2f\n", media);
    printf("O maior elemento é: %.2f\n", maior);
    printf("O menor elemento é: %.2f\n", menor);
    printf("A quantidade de elementos positivos é: %d\n", positivos);
    printf("A quantidade de elementos negativos é: %d\n", negativos);

    return 0;
}
```

4 Escreva um programa que [Entrada]: receba do usuário: 1) o valor base de indenização por um plano de saúde; 2) a idade do paciente indenizado e; 3) o nome completo do paciente. [Processamento]: Seu programa deve calcular o reajuste de indenização sobre o valor base de acordo com a idade do paciente. Seu programa deve aumentar o valor da indenização em: 30% se o paciente tiver até 12 anos; 10% se o paciente tiver idade entre 13 e 49 anos (inclusive); 15% se o paciente tiver idade entre 50 e 65 anos (inclusive) e; 35% se o paciente tiver idade acima de 65 anos. [Saída]: AO FINAL, SOMENTE AO FINAL, Seu programe deve mostrar (um paciente por linha): o nome do paciente, a idade do paciente e qual foi o valor da indenização reajustado. Obs: o usuário deve fazer esse procedimento para quantos clientes ELE QUISE. Dica: voce PODE usar três vetores: um para os nomes, outro para as idades e outro para as indenizações reajustadas

```
#include <stdio.h>
#include <stdlib.h>
#define MAX_PACIENTES 100
int main() {
    float valor_base, indenizacoes[MAX_PACIENTES];
    int idade, idades[MAX_PACIENTES];
    char nomes[MAX_PACIENTES][50];
    int i = 0;
    printf("Digite o valor base de indenização:\n");
    scanf("%f", &valor_base);
    while(1) {
        printf("Digite o nome do paciente ou 'sair' para terminar:\n");
        scanf("%s", nomes[i]);
        if(strcmp(nomes[i], "sair") == 0) break;
        printf("Digite a idade do paciente:\n");
        scanf("%d", &idade);
        if(idade <= 12) {
            indenizacoes[i] = valor_base * 1.3;
        } else if(idade >= 13 && idade <= 49) {
            indenizacoes[i] = valor_base * 1.1;
        } else if(idade >= 50 && idade <= 65) {
            indenizacoes[i] = valor_base * 1.15;
        } else {
            indenizacoes[i] = valor_base * 1.35;
        }
        idades[i] = idade;
        i++;
    }
    for(int j = 0; j < i; j++) {
        printf("Paciente: %s, Idade: %d, Indenização: %.2f\n", nomes[j], idades[j], indenizacoes[j]);
    }
    return 0;
}
```

5 Escreva um programa que leia três vetores do mesmo tamanho (N) e:

- a) calcule a norma de cada um dos vetores, informe qual vetor tem a maior norma1 ;
- b) calcule o vetor soma dos três vetores.

```
#include <stdio.h>#include <math.h>#define N 3
float norma(float vetor[N]) {
    float soma = 0;
    for(int i = 0; i < N; i++) {
        soma += vetor[i] * vetor[i];    } }
int main() {
    float vetor1[N], vetor2[N], vetor3[N], soma[N];
    float norma1, norma2, norma3;
    printf("Digite os elementos do vetor 1:\n");
    for(int i = 0; i < N; i++) {
        scanf("%f", &vetor1[i]);    }
    printf("Digite os elementos do vetor 2:\n");
    for(int i = 0; i < N; i++) {
        scanf("%f", &vetor2[i]);    }
    printf("Digite os elementos do vetor 3:\n");
    for(int i = 0; i < N; i++) {
        scanf("%f", &vetor3[i]);    }
    norma1 = norma(vetor1);
    norma2 = norma(vetor2);
    norma3 = norma(vetor3);
    printf("Norma do vetor 1: %.2f\n", norma1);
    printf("Norma do vetor 2: %.2f\n", norma2);
    printf("Norma do vetor 3: %.2f\n", norma3);
    if(norma1 > norma2 && norma1 > norma3) {
        printf("O vetor 1 tem a maior norma.\n");
    } else if(norma2 > norma1 && norma2 > norma3) {
        printf("O vetor 2 tem a maior norma.\n");
    } else {
        printf("O vetor 3 tem a maior norma.\n");
    }
    for(int i = 0; i < N; i++) {
        soma[i] = vetor1[i] + vetor2[i] + vetor3[i];    }
    printf("Vetor soma: ");
    for(int i = 0; i < N; i++) {
        printf("%.2f ", soma[i]);
    }
    printf("\n");    return sqrt(soma);
    return 0;
}
```

6 Escreva um programa que faça um cadastro de clientes. Seu programa deve [Entrada]: receber os seguintes dados do usuário: 1) o nome completo; 2) o RG do cliente; 3) o CPF e; 4) o telefone do cliente. [Processamento]: Seu programa deve armazenar todos os dados em uma ÚNICA LISTA. [Saída]: AO FINAL, SOMENTE AO FINAL, Seu programe deve mostrar (um cliente por linha): a) o nome completo do paciente, b) O RG; c) o CPF e; d) o telefone do cliente. Obs: o usuário deve fazer esse procedimento para quantos clientes ELE QUISER.

```
#include <stdio.h>
#include <string.h>

#define MAX_CLIENTES 100
#define MAX_NOME 50
#define MAX_RG 15
#define MAX_CPF 15
#define MAX_TELEFONE 15

typedef struct {
    char nome[MAX_NOME];
    char rg[MAX_RG];
    char cpf[MAX_CPF];
    char telefone[MAX_TELEFONE];
} Cliente;

int main() {
    Cliente clientes[MAX_CLIENTES];
    int i = 0;

    while(1) {
        printf("Digite o nome completo do cliente ou 'sair' para terminar:\n");
        scanf("%s", clientes[i].nome);
        if(strcmp(clientes[i].nome, "sair") == 0) break;

        printf("Digite o RG do cliente:\n");
        scanf("%s", clientes[i].rg);

        printf("Digite o CPF do cliente:\n");
        scanf("%s", clientes[i].cpf);

        printf("Digite o telefone do cliente:\n");
        scanf("%s", clientes[i].telefone);

        i++;
    }

    for(int j = 0; j < i; j++) {
        printf("Cliente: %s, RG: %s, CPF: %s, Telefone: %s\n", clientes[j].nome, clientes[j].rg, clientes[j].cpf, clientes[j].telefone);
    }

    return 0;
}
```

7 Escreva um programa que leia e mostre uma lista de 10 elementos inteiros. Em seguida, conte quantos valores pares existem na lista, por fim, exiba a quantidade na tela.

```
#include <stdio.h>

int main() {
    int numeros[10], pares = 0;

    printf("Digite 10 números inteiros:\n");
    for(int i = 0; i < 10; i++) {
        scanf("%d", &numeros[i]);
        if(numeros[i] % 2 == 0) {
            pares++;
        }
    }

    printf("Os números digitados foram:\n");
    for(int i = 0; i < 10; i++) {
        printf("%d ", numeros[i]);
    }
    printf("\n");

    printf("A quantidade de números pares é: %d\n", pares);

    return 0;
}
```

8 Elaborar um programa que lê um conjunto de vários valores inteiros e os coloca em 2 listas conforme forem pares ou ímpares (uma lista para números pares e outra lista para números ímpares). O tamanho da lista é de 5 posições. Quando alguma lista estiver cheia, ela deverá ser exibida na tela. Cada lista poderá ser preenchida tantas vezes quantas forem necessárias. Caso a leitura seja finalizada sem o preenchimento completo de alguma das lista, as listas deverão ser exibidas na tela.

```
#include <stdio.h>
#define TAMANHO 5
void exibir_lista(int lista[], int tamanho) {
    for(int i = 0; i < tamanho; i++) {
        printf("%d ", lista[i]);
    }
    printf("\n");
}
int main() {
    int pares[TAMANHO], impares[TAMANHO];
    int contador_pares = 0, contador_impares = 0;
    int numero;
    while(1) {
        printf("Digite um número inteiro ou -1 para terminar:\n");
        scanf("%d", &numero);
        if(numero == -1) break;
        if(numero % 2 == 0) {
            pares[contador_pares] = numero;
            contador_pares++;
            if(contador_pares == TAMANHO) {
                printf("Lista de números pares:\n");
                exibir_lista(pares, TAMANHO);
                contador_pares = 0;
            }
        } else {
            impares[contador_impares] = numero;
            contador_impares++;
            if(contador_impares == TAMANHO) {
                printf("Lista de números ímpares:\n");
                exibir_lista(impares, TAMANHO);
                contador_impares = 0;
            }
        }
    }
    if(contador_pares > 0) {
        printf("Lista de números pares:\n");
        exibir_lista(pares, contador_pares);
    }
    if(contador_impares > 0) {
        printf("Lista de números ímpares:\n");
        exibir_lista(impares, contador_impares);
    }
    return 0;
}
```

9 Escreva um programa que leia uma lista G de 20 elementos caractere que representa o gabarito de uma prova com 20 questões. A seguir, para cada um dos 50 alunos de uma turma, leia o nome e o vetor de respostas (R) do aluno, após ler conte o número de acertos desse aluno e mostre o numero de acertos e uma mensagem APROVADO, se a nota FINAL for maior ou igual a 6 ou mostre o numeros de acertos e uma mensagem de REPROVADO, caso contrário. Cada questão correta vale 0,5 ponto.

```
#include <stdio.h>
#include <string.h>

#define NUM_QUESTOES 20
#define NUM_ALUNOS 50
#define MAX_NOME 50

int main() {
    char gabarito[NUM_QUESTOES + 1], respostas[NUM_QUESTOES + 1];
    char nome[MAX_NOME];
    int acertos;
    float nota;

    printf("Digite o gabarito da prova:\n");
    scanf("%s", gabarito);

    for(int i = 0; i < NUM_ALUNOS; i++) {
        printf("Digite o nome do aluno:\n");
        scanf(" %[^\\n]", nome);

        printf("Digite as respostas do aluno:\n");
        scanf("%s", respostas);

        acertos = 0;
        for(int j = 0; j < NUM_QUESTOES; j++) {
            if(gabarito[j] == respostas[j]) {
                acertos++;
            }
        }

        nota = acertos * 0.5;

        printf("Aluno: %s, Acertos: %d, Nota: %.1f, ", nome, acertos, nota);
        if(nota >= 6) {
            printf("APROVADO\\n");
        } else {
            printf("REPROVADO\\n");
        }
    }

    return 0;
}
```

10. Escreva um programa que leia uma lista G de 20 elementos caractere que representa o gabarito de uma prova com 20 questões. A seguir, para cada um dos 50 alunos de uma turma, leia o nome e o vetor de respostas (R) do aluno, após ler conte o número de acertos e ARMAZENE o nome e a NOTA FINAL desse aluno. APÓS processar a contagem para os 50 alunos, mostre o nome do aluno e a NOTA FINAL seguida da mensagem APROVADO, se a nota FINAL for maior ou igual a 6 ou seguida da mensagem REPROVADO, caso contrário. Cada questão correta vale 0,5 ponto.

```
#define MAX_NOME 50

typedef struct {
    char nome[MAX_NOME];
    float nota;
} Aluno;

int main() {
    char gabarito[NUM_QUESTOES + 1], respostas[NUM_QUESTOES + 1];
    Aluno alunos[NUM_ALUNOS];
    int acertos;

    printf("Digite o gabarito da prova:\n");
    scanf("%s", gabarito);

    for(int i = 0; i < NUM_ALUNOS; i++) {
        printf("Digite o nome do aluno:\n");
        scanf(" %[^\\n]", alunos[i].nome);

        printf("Digite as respostas do aluno:\n");
        scanf("%s", respostas);

        acertos = 0;
        for(int j = 0; j < NUM_QUESTOES; j++) {
            if(gabarito[j] == respostas[j]) {
                acertos++;
            }
        }

        alunos[i].nota = acertos * 0.5;
    }

    for(int i = 0; i < NUM_ALUNOS; i++) {
        printf("Aluno: %s, Nota: %.1f, ", alunos[i].nome, alunos[i].nota);
        if(alunos[i].nota >= 6) {
            printf("APROVADO\\n");
        } else {
            printf("REPROVADO\\n");
        }
    }

    return 0;
}
```


11 Faça um programa que leia um código numérico inteiro e uma lista de 15 posições de números reais. Se o código for zero, termine o programa. Se o código for 1, mostre a lista na ordem direta. Se o código for 2, mostre a lista na ordem inversa.

```
#include <stdio.h>

#define TAMANHO 15

int main() {
    char lista[TAMANHO];
    int codigo;

    printf("Digite os elementos da lista:\n");
    for(int i = 0; i < TAMANHO; i++) {
        scanf(" %c", &lista[i]);
    }

    while(1) {
        printf("Digite um código (0 para terminar, 1 para ordem direta, 2 para ordem inversa):\n");
        scanf("%d", &codigo);

        if(codigo == 0) {
            break;
        } else if(codigo == 1) {
            printf("Lista na ordem direta:\n");
            for(int i = 0; i < TAMANHO; i++) {
                printf("%c ", lista[i]);
            }
            printf("\n");
        } else if(codigo == 2) {
            printf("Lista na ordem inversa:\n");
            for(int i = TAMANHO - 1; i >= 0; i--) {
                printf("%c ", lista[i]);
            }
            printf("\n");
        } else {
            printf("Código inválido.\n");
        }
    }

    return 0;
}
```

12 Construa um programa que leia dois números inteiros a e b, uma lista de tamanho N e exiba como resposta a contagem de quantos elementos do vetor estão no intervalo fechado [a; b].

```
#include <stdio.h>

#define N 10

int main() {
    int a, b, lista[N], contagem = 0;

    printf("Digite os valores de a e b:\n");
    scanf("%d %d", &a, &b);

    printf("Digite os elementos da lista:\n");
    for(int i = 0; i < N; i++) {
        scanf("%d", &lista[i]);
        if(lista[i] >= a && lista[i] <= b) {
            contagem++;
        }
    }

    printf("A quantidade de elementos no intervalo [%d, %d] é: %d\n", a, b, contagem);

    return 0;
}
```