

Exercício de Programação Estruturada 6

Exercício de Programação Estruturada 6

1.

```
#include <stdio.h>

void fillVector(float vet[], int size)
{
    int i;
    for (i = 0; i < size; i++)
    {
        printf("Digite o valor do mês %d: ", i + 1);
        scanf(" %f", &vet[i]);
    }
}

void calculateProfit(float vetGanhos[], float vetGastos[], float
vetLucros[], int months)
{
    int i;
    for (i = 0; i < months; i++)
    {
        vetLucros[i] = vetGanhos[i] - vetGastos[i];
    }
}

void printVector(float vet[], int size)
{
    int i;
    for (i = 0; i < size; i++)
    {
        printf("%.2f\n", vet[i]);
    }
}

int main()
{
    int months;
    months = 12;
    float vetGanhos[months], vetGastos[months], vetLucros[months];
    printf("Preencha os ganhos da empresa em cada mês do ano de 2022:\n");
```

```

    fillVector(vetGanhos, months);
    printf("Preencha os gastos da empresa em cada mês do ano de 2022:\n");
    fillVector(vetGastos, months);
    calculateProfit(vetGanhos, vetGastos, vetLucros, months);
    printf("Vetor de ganhos por mês:\n");
    printVector(vetGanhos, months);
    printf("Vetor de gastos por mês:\n");
    printVector(vetGastos, months);
    printf("Vetor de lucros por mês:\n");
    printVector(vetLucros, months);
    return 0;
}

```

2.

```

#include <stdio.h>

void fillVector(float vet[], int size)
{
    int i;
    for (i = 0; i < size; i++)
    {
        printf("Digite o valor do mês %d: ", i + 1);
        scanf(" %f", &vet[i]);
    }
}

void calculateProfit(float vetGanhos[], float vetGastos[], float
vetLucros[], int months)
{
    int i;
    for (i = 0; i < months; i++)
    {
        vetLucros[i] = vetGanhos[i] - vetGastos[i];
    }
}

void printVector(float vet[], int size)
{
    int i;
    for (i = 0; i < size; i++)
    {
        printf("%.2f\n", vet[i]);
    }
}

```

```

float sumVector(float vet[], int size)
{
    int i;
    float sum;
    sum = 0;
    for (i = 0; i < size; i++)
    {
        sum = sum + vet[i];
    }
    return sum;
}

int main()
{
    int months;
    months = 12;
    float vetGanhos[months], vetGastos[months], vetLucros[months],
totalGanhos, totalGastos, totalLucros;
    printf("Preencha os ganhos da empresa em cada mês do ano de 2022:\n");
    fillVector(vetGanhos, months);
    printf("Preencha os gastos da empresa em cada mês do ano de 2022:\n");
    fillVector(vetGastos, months);
    calculateProfit(vetGanhos, vetGastos, vetLucros, months);
    totalGanhos = sumVector(vetGanhos, months);
    totalGastos = sumVector(vetGastos, months);
    totalLucros = sumVector(vetLucros, months);
    printf("Vetor de ganhos por mês:\n");
    printVector(vetGanhos, months);
    printf("O total de ganhos no ano é de %.2f.\n", totalGanhos);
    printf("Vetor de gastos por mês:\n");
    printVector(vetGastos, months);
    printf("O total de gastos no ano é de %.2f.\n", totalGastos);
    printf("Vetor de lucros por mês:\n");
    printVector(vetLucros, months);
    printf("O total de lucro no ano é de %.2f.\n", totalLucros);

    return 0;
}

```

3.

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

```
int main()
```

```

{
    int voto, totais[10], i;
    float percJoao, percMaria, percPedro, percLuis, percAna, percLuiza,
    percSilvia, percAndre, percBranco, percNulo;
    char answer;
    for (i = 0; i < 10; i++)
    {
        totais[i] = 0;
    }
    while (1==1)
    {
        answer = '0';
        /*totais[9] = nulo, totais[10] = total de eleitores*/
        printf("0 - Voto em branco\n1 - Voto em João\n2 - Voto em Maria\n3 -
Voto em Pedro\n4 - Voto em Luís\n5 - Voto em Ana\n6 - Voto em Luiza\n7 -
Voto em Silvia\n8 - Voto em André\nDigite seu voto:");
        scanf("%d", &voto);
        totais[10] = totais[10] + 1;
        switch (voto)
        {
            case 0:
                totais[0] = totais[0] + 1;
                break;
            case 1:
                totais[1] = totais[1] + 1;
                break;
            case 2:
                totais[2] = totais[2] + 1;
                break;
            case 3:
                totais[3] = totais[3] + 1;
                break;
            case 4:
                totais[4] = totais[4] + 1;
                break;
            case 5:
                totais[5] = totais[5] + 1;
                break;
            case 6:
                totais[6] = totais[6] + 1;
                break;
            case 7:
                totais[7] = totais[7] + 1;

```

```

        break;
    case 8:
        totais[8] = totais[8] + 1;
        break;
    default:
        totais[9] = totais[9] + 1;
        break;
}
printf("Outro eleitor deseja votar? [S/N]\n");
while (1==1)
{
    scanf("%c", &answer);
    if (answer == 'S' || answer == 's' || answer == 'N' || answer ==
'n')
    {
        break;
    }
    else
    {
        printf("O usuário deve responder s ou S, para sim ou n ou N,
para não.\n");
    }

}
if (answer == 'N' || answer == 'n')
{
    break;
}
}
percJoao = (float)totais[1] / totais[10] * 100;
percMaria = (float)totais[2] / totais[10] * 100;
percPedro = (float)totais[3] / totais[10] * 100;
percLuis = (float)totais[4] / totais[10] * 100;
percAna = (float)totais[5] / totais[10] * 100;
percLuiza = (float)totais[6] / totais[10] * 100;
percSilvia = (float)totais[7] / totais[10] * 100;
percAndre = (float)totais[8] / totais[10] * 100;
percBranco = (float)totais[0] / totais[10] * 100;
percNulo = (float)totais[9] / totais[10] * 100;
printf("\nPercentual de votos:\n");
printf("João: %.2f%%\n", percJoao);
printf("Maria: %.2f%%\n", percMaria);
printf("Pedro: %.2f%%\n", percPedro);

```

```

printf("Luís: %.2f%%\n", percLuis);
printf("Ana: %.2f%%\n", percAna);
printf("Luiza: %.2f%%\n", percLuiza);
printf("Silvia: %.2f%%\n", percSilvia);
printf("André: %.2f%%\n", percAndre);
printf("Voto em Branco: %.2f%%\n", percBranco);
printf("Voto Nulo: %.2f%%\n", percNulo);
return 0;
}

```

4.

```

#include <stdio.h>
#include <stdlib.h>

void sortearLado(int *resultado) {
    int lado;
    lado = (rand() % 6) + 1;
}

void realizarSimulacao(int *resultado, int numLancamentos) {
    int i;
    for (i = 0; i < numLancamentos; i++) {
        sortearLado(resultado);
    }
}

void exibirResultado(int *resultado) {
    int i;
    printf("Resultado da simulação:\n");
    for (i = 0; i < 6; i++) {
        printf("Lado %d: %d vezes\n", i + 1, resultado[i]);
    }
}

int main() {
    int resultado[6], numLancamentos, i;
    numLancamentos = 1000000;
    for (i = 0; i < 10; i++)
    {
        resultado[i] = 0;
    }
    srand(1);
}

```

```
    realizarSimulacao(resultado, numLancamentos);  
    exibirResultado(resultado);  
  
    return 0;  
}
```

5.

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

```
void inserirValor(int vetor[], int *tamanho, int *posicoesPreenchidas, int  
valor) {  
    if (*posicoesPreenchidas < *tamanho) {  
        vetor[*posicoesPreenchidas] = valor;  
        *posicoesPreenchidas = *posicoesPreenchidas + 1;  
        printf("Valor inserido com sucesso.\n");  
    } else {  
        printf("O vetor está cheio. Não é possível inserir mais  
valores.\n");  
    }  
}
```

```
void buscarValor(int vetor[], int posicoesPreenchidas, int valor) {  
    int encontrado = 0;  
  
    printf("Posições onde o valor foi encontrado: ");  
    for (int i = 0; i < posicoesPreenchidas; i++) {  
        if (vetor[i] == valor) {  
            printf("%d ", i);  
            encontrado = 1;  
        }  
    }  
  
    if (!encontrado) {  
        printf("Valor não encontrado no vetor.\n");  
    }  
}
```

```
void exibirVetor(int vetor[], int posicoesPreenchidas) {  
    printf("Valores no vetor: ");  
    for (int i = 0; i < posicoesPreenchidas; i++) {  
        printf("%d ", vetor[i]);  
    }  
    printf("\n");  
}
```

```

}

int main() {
    int tamanhoMaximo = 50;
    int vetorA[tamanhoMaximo], vetorB[tamanhoMaximo];
    int tamanhoA = tamanhoMaximo, tamanhoB = tamanhoMaximo;
    int posicoesPreenchidasA = 0, posicoesPreenchidasB = 0;
    int opcao, valor;

    while (1==1) {
        printf("\nMenu Principal\nOpções:\n1 - Inserir novo valor\n2 - Procurar valor\n3 - Exibir vetor\n0 - Encerrar programa\nInforme a opção desejada: ");
        scanf("%d", &opcao);

        switch (opcao) {
            case 1:
                printf("Informe o valor a ser inserido: ");
                scanf("%d", &valor);

                if (valor % 2 == 0) {
                    inserirValor(vetorA, &tamanhoA, &posicoesPreenchidasA,
valor);
                } else {
                    inserirValor(vetorB, &tamanhoB, &posicoesPreenchidasB,
valor);
                }
                break;
            case 2:
                printf("Informe o valor a ser procurado: ");
                scanf("%d", &valor);

                if (valor % 2 == 0) {
                    buscarValor(vetorA, posicoesPreenchidasA, valor);
                } else {
                    buscarValor(vetorB, posicoesPreenchidasB, valor);
                }
                break;
            case 3:
                printf("Informe o vetor a ser exibido (A=1, B=2): ");
                scanf("%d", &valor);
                if (valor == 1) {
                    exibirVetor(vetorA, posicoesPreenchidasA);
                }
            }
        }
    }
}

```



```

        } else if (valor == 2) {
            exibirVetor(vetorB, posicoesPreenchidasB);
        } else {
            printf("Opção inválida.\n");
        }
        break;
    case 0:
        return 0;
    default:
        printf("Opção inválida.\n");
    }
}

return 0;
}

```

6.

```

#include <stdio.h>

void inserirValor(float vetor[], int tamanho, int *posicoesPreenchidas,
float valor) {
    if (*posicoesPreenchidas < tamanho) {
        for (int i = 0; i < *posicoesPreenchidas; i++) {
            if (vetor[i] == valor) {
                printf("Valor já existe no vetor. Inserção cancelada.\n");
                return;
            }
        }

        vetor[*posicoesPreenchidas] = valor;
        *posicoesPreenchidas = *posicoesPreenchidas + 1;
        printf("Valor inserido com sucesso.\n");
    } else {
        printf("O vetor está cheio. Não é possível inserir mais
valores.\n");
    }
}

int buscarValor(float vetor[], int posicoesPreenchidas, float valor) {
    for (int i = 0; i < posicoesPreenchidas; i++) {
        if (vetor[i] == valor) {
            return i;
        }
    }
}

```

```

    }
    return -1;
    /*-1 significa que o valor não encontrado*/
}

void exibirVetor(float vetor[], int posicoesPreenchidas) {
    printf("Valores no vetor: ");
    for (int i = 0; i < posicoesPreenchidas; i++) {
        printf("%.2f ", vetor[i]);
    }
    printf("\n");
}

void removerValor(float vetor[], int tamanho, int *posicoesPreenchidas,
float valor) {
    int posicao = buscarValor(vetor, *posicoesPreenchidas, valor);
    if (posicao != -1) {
        for (int i = posicao; i < *posicoesPreenchidas - 1; i++) {
            vetor[i] = vetor[i + 1];
        }
        (*posicoesPreenchidas)--;
        printf("Valor removido com sucesso.\n");
    } else {
        printf("Valor não encontrado no vetor. Remoção cancelada.\n");
    }
}

int main() {
    int tamanhoMaximo = 100;
    float vetorA[tamanhoMaximo], vetorB[tamanhoMaximo];
    int posicoesPreenchidasA = 0, posicoesPreenchidasB = 0;
    int opcao;
    float valor;

    while (1) {
        printf("\nMenu Principal\nOpções:\n1 - Inserir novo valor\n2 -
Procurar valor\n3 - Exibir vetor\n4 - Remover valor\n0 - Encerrar
programa\nInforme a opção desejada: ");
        scanf("%d", &opcao);

        switch (opcao) {
            case 1:
                printf("Informe o valor a ser inserido: ");

```

```

scanf("%f", &valor);

if (valor == (int)valor) {
    if ((int)valor % 2 == 0) {
        inserirValor(vetorA, tamanhoMaximo,
&posicoesPreenchidasA, valor);
    } else {
        inserirValor(vetorB, tamanhoMaximo,
&posicoesPreenchidasB, valor);
    }
} else {
    printf("Valor deve ser um número inteiro.\n");
}
break;
case 2:
    printf("Informe o valor a ser procurado: ");
    scanf("%f", &valor);

    if (valor == (int)valor) {
        int posicaoA = buscarValor(vetorA, posicoesPreenchidasA,
valor);

        int posicaoB = buscarValor(vetorB, posicoesPreenchidasB,
valor);

        if (posicaoA != -1) {
            printf("Valor encontrado no vetor A na posição
%d.\n", posicaoA);
        } else if (posicaoB != -1) {
            printf("Valor encontrado no vetor B na posição
%d.\n", posicaoB);
        } else {
            printf("Valor não encontrado em nenhum vetor.\n");
        }
    } else {
        printf("Valor deve ser um número inteiro.\n");
    }
    break;
case 3:
    printf("Informe o vetor a ser exibido (A=1, B=2): ");
    scanf("%d", &opcao);
    if (opcao == 1) {
        exibirVetor(vetorA, posicoesPreenchidasA);
    } else if (opcao == 2) {

```

```
        exibirVetor(vetorB, posicoesPreenchidasB);
    } else {
        printf("Opção inválida.\n");
    }
    break;
case 4:
    printf("Informe o valor a ser removido: ");
    scanf("%f", &valor);

    if (valor == (int)valor) {
        if ((int)valor % 2 == 0) {
            removerValor(vetorA, tamanhoMaximo,
&posicoesPreenchidasA, valor);
        } else {
            removerValor(vetorB, tamanhoMaximo,
&posicoesPreenchidasB, valor);
        }
    } else {
        printf("Valor deve ser um número inteiro.\n");
    }
    break;
case 0:
    return 0;
default:
    printf("Opção inválida.\n");
}
}

return 0;
}
```