



Disciplina	Curso	Turno	Período
Algoritmos e Estruturas de Dados I	Ciência da Computação	Manhã	1º
Professores	Felipe Cunha		

Prova 02 – Exemplo

1. Considere o programa abaixo:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 void modulo1 (int* vetor, int N){
4     for (int pos=0; pos < N; pos++){
5         printf("Digite: ");
6         scanf("%d", &(vetor+pos));
7     }
8 }
9 void modulo2 (int* dados, int N){
10    printf("Conteúdo: ");
11    for (int i=0; i < N; i++) {
12        printf("%d ",(dados+i));
13    }
14 }
15 int main(void) {
16     int total = 10;
17     int valores[total];
18     modulo1(&valores, total);
19     modulo2(&valores, total);
20     return 0;
21 }
```

Tarefas (5 pontos):

- Reescreva apenas as linhas incorretas no trecho de código a seguir de tal forma que seja preenchido e exibido o conteúdo de um vetor de 10 inteiros utilizando ponteiro para indicar o vetor e aritmética de ponteiro para acessar cada elemento.
- Explique cada uma das correções realizadas. **A organização dos módulos deve ser mantida.**

RESPOSTA:

Linhas corrigidas:

```
1     scanf ("%d", (vetor+pos));
2     ...
3     printf ("%d ",*(dados+i));
4     ...
5     modulo1(valores,total);
6     modulo2(valores,total);
```

Explicações:

- Na linha do scanf, o operador & foi removido, pois o ponteiro vetor já aponta para o endereço do primeiro elemento do vetor, e a aritmética de ponteiros permite acessar o endereço correto somando o deslocamento pos.
- Na linha do printf, o operador de desreferenciação * foi adicionado para acessar o valor armazenado no endereço calculado por (dados+i).
- Nas chamadas das funções modulo1 e modulo2, o operador & foi removido, pois o nome do array valores já representa o endereço do primeiro elemento do array, e não é necessário passar o endereço do array novamente.

2. (5 pontos) Elabore um trecho de código que leia uma frase e um caracter, trocando na frase o caracter lido por * (caracter asterisco).

RESPOSTA:

```
1 #include <stdio.h>
2 #include <string.h>
3 int main() {
4     char frase[100];
5     char caractere;
6
7     printf("Digite uma frase: ");
8     fgets(frase, sizeof(frase), stdin);
9
10    printf("Digite um caracter para substituir por
11        '*': ");
12    scanf(" %c", &caractere);
13
14    for (int i = 0; i < strlen(frase); i++) {
15        if (frase[i] == caractere) {
16            frase[i] = '*';
17        }
18    }
19
20    printf("Frase modificada: %s", frase);
21    return 0;
22 }
```

3. (15 pontos) Faça um programa em C padrão que realiza as seguintes tarefas.

- (a) (5 pontos) Crie um **procedimento** que aloque dinamicamente uma matriz de números reais com tamanho 10×10 . Os valores devem ser lidos de um **arquivo de texto**, cujo nome será digitado pelo usuário no teclado.

RESPOSTA:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 void preencherMatriz(char* nomeArquivo, float** matriz, int linhas, int colunas) {
4     FILE* arquivo = fopen(nomeArquivo, "r");
5     if (arquivo == NULL) {
6         printf("Erro ao abrir o arquivo.\n");
7         return;
8     }
9     for (int i = 0; i < linhas; i++) {
10        for (int j = 0; j < colunas; j++) {
11            fscanf(arquivo, "%f", &matriz[i][j]);
12        }
13    }
14    fclose(arquivo);
15 }
```

- (b) (5 pontos) Crie um **procedimento** que percorra toda a matriz e retorne um **vetor** contendo a **diagonal principal**.

RESPOSTA:

```
1 void obterDiagonalPrincipal(float** matriz, float * diagonal, int tamanho) {
2     for (int i = 0; i < tamanho; i++) {
3         diagonal[i] = matriz[i][i];
4     }
5 }
```

- (c) (5 pontos) No programa principal (**main**), siga os seguintes passos:

- i. Peça o nome do arquivo e chame o procedimento para preencher a matriz.
- ii. Chame a função que retorna o vetor e mostre os valores do vetor na tela.
- iii. Exiba a matriz e o vetor final.

RESPOSTA:

```
1 int main() {
2     int linhas = 10, colunas = 10;
3     float matriz[linhas][colunas];
4     float diagonal[10];
5     char nomeArquivo[50];
6
7     printf("Digite o nome do arquivo: ");
8     scanf("%s", nomeArquivo);
9
10    preencherMatriz(nomeArquivo, matriz, linhas,
11                      colunas);
12
13    obterDiagonalPrincipal(matriz, linhas);
14
15    printf("Matriz:\n");
16    for (int i = 0; i < linhas; i++) {
17        for (int j = 0; j < colunas; j++) {
18            printf("%.2f ", matriz[i][j]);
19        }
20        printf("\n");
21    }
22
23    printf("Diagonal Principal:\n");
24    for (int i = 0; i < linhas; i++) {
25        printf("%.2f ", diagonal[i]);
26    }
27    return 0;
}
```