



PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS

Instituto de Ciências Exatas e Informática

Departamento de Ciência da Computação

PUC Minas

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

Prova 02

1. Considere o programa abaixo:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 void preencher(float* v, int n){
5     for (int i = 0; i <= n; i++){
6         printf("Valor %d: ", i);
7         scanf("%f", &(v + i));
8     }
9 }
10 void dobrar(float* origem, float* destino, int n){
11     for (int k = 0; k < n; k++){
12         *(destino + k) = *(origem + k) * 2.0f;
13     }
14 }
15 void mostrar(float* v, int n){
16     printf("Vetor: ");
17     for (int j = 0; j < n; j++){
18         printf("%d ", *(v + j));
19     }
20 }
21 int main(void){
22     int N = 8;
23     float a[N];
24     float* b = NULL;
25
26     preencher(&a, N);
27     dobrar(a, b, N);
28     mostrar(b, N);
29
30     return 0;
31 }
```

Tarefas (5 pontos):

- (a) Reescreva apenas as linhas incorretas do trecho de código acima de modo que o programa tenha a sua execução correta, preenchendo e exibindo o conteúdo de um vetor de 8 números reais utilizando ponteiro para indicar o vetor e aritmética de ponteiro para acessar cada elemento.
- (b) Explique cada uma das correções realizadas (por exemplo: ajuste de limites do laço, endereço passado ao `scanf`, especificador de formato, tamanho da alocação e passagem de ponteiro ao módulo).
- (c) **ATENÇÃO!!! A organização dos módulos (preencher, dobrar, mostrar) deve ser mantida.**

RESPOSTA:

```
1 void preencher(float* v, int n){  
2     for (int i = 0; i < n; i++){ // Limite do laço  
3         printf("Valor %d: ", i);  
4         scanf("%f", (v + i)); // Endereço passado ao  
        scanf  
5     }  
6 }  
7 void dobrar(float* origem, float* destino, int n){  
8     // esta correto  
9 }  
10 void mostrar(float* v, int n){  
11     printf("Vetor: ");  
12     for (int j = 0; j < n; j++){  
13         printf("%.2f ", *(v + j)); // Especificador de  
            formato  
14     }  
15 }  
16 int main(void){  
17     int N = 8;  
18     float a[N];  
19     float b[N]; // Alocação do vetor  
20     preencher(a, N); // Passagem do vetor  
21     dobrar(a, b, N);  
22     mostrar(b, N);  
23     return 0;  
24 }
```

RESPOSTA:

Explicações das correções:

- No módulo preencher, o laço foi corrigido para iterar até $i < n$ para evitar acessar um índice fora dos limites do vetor.
- No scanf, o endereço foi corrigido para $(v + i)$ para passar o endereço correto do elemento do vetor.
- No módulo dobrar, o destino não foi alocado como vetor. Por esse motivo, foi corrigida a declaração do vetor no int main para reservar o espaço para o vetor em memória.
- No módulo mostrar, o especificador de formato foi corrigido para %.2f para exibir números reais com duas casas decimais.
- No main, a passagem do vetor para o módulo preencher foi corrigida para a em vez de &a, pois o nome do vetor já representa o endereço do primeiro elemento.

2. (5 pontos) Faça um programa em C que siga os passos abaixo com atenção:

- (a) Peça ao usuário para **digitar uma linha de texto**.
- (b) Em seguida, peça ao usuário para **digitar uma palavra** que será procurada nessa linha.
- (c) O programa deve **verificar se a palavra aparece dentro da linha**. Caso a palavra exista, exiba na tela a **posição inicial** onde ela foi encontrada.
- (d) Caso a palavra não exista na linha, exiba uma mensagem informando: “Palavra não encontrada na linha.”

RESPOSTA:

```
1 #include <stdio.h>
2 #include <string.h>
3
4 int main() {
5     char linha[200];
6     char palavra[50];
7     int i, j, encontrado;
8
9     printf("Digite uma linha de texto:\n");
10    fgets(linha, 200, stdin);
11    linha[strcspn(linha, "\n")] = '\0';
12
13    printf("Digite a palavra que deseja procurar:\n");
14    scanf("%s", palavra);
15
16    int tamLinha = strlen(linha);
17    int tamPalavra = strlen(palavra);
18    int posicao = -1;
19
20    for (i = 0; i <= tamLinha - tamPalavra; i++) {
21        encontrado = 1;
22        for (j = 0; j < tamPalavra; j++) {
23            if (linha[i + j] != palavra[j]) {
24                encontrado = 0;
25                j = tamPalavra;
26            }
27        }
28        if (encontrado) {
29            posicao = i;
30            i = tamLinha;
31        }
32    }
33    if (posicao != -1)
34        printf("Palavra encontrada na posicao: %d\n",
35               posicao);
36    else
37        printf("Palavra nao encontrada na linha.\n");
38
39 }
```

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

- (a) (5 pontos) Implemente um **procedimento** que receba, como parâmetro, uma matriz de números reais de dimensão 25×25 e a preencha com valores lidos de um **arquivo de texto**. O nome do arquivo deve ser informado pelo usuário por meio do teclado.

RESPOSTA:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 void preencher_matriz(float matriz[25][25], char*
4     nome_arquivo) {
5     FILE* arquivo = fopen(nome_arquivo, "r");
6     if (arquivo == NULL) {
7         printf("Erro ao abrir o arquivo.\n");
8         return;
9     }
10    for (int i = 0; i < 25; i++) {
11        for (int j = 0; j < 25; j++) {
12            fscanf(arquivo, "%f", &matriz[i][j]);
13        }
14    }
15 }
```

- (b) (5 pontos) Implemente uma **função recursiva** que percorra todos os elementos de uma coluna de uma matriz e retorne o **menor valor encontrado nessa coluna**. Em seguida, crie um **vetor** para armazenar os menores valores de cada coluna da matriz, preenchendo-o com os valores retornados pela função.

RESPOSTA:

```
1 float encontrar_menor_valor(float matriz[25][25],
2     int coluna, int linha, int total_linhas) {
3     if (linha == total_linhas - 1) {
4         return matriz[linha][coluna];
5     }
6     float menor_restante =
7         encontrar_menor_valor(matriz, coluna, linha
8             + 1, total_linhas);
9     return (matriz[linha][coluna] <
10         menor_restante) ? matriz[linha][coluna] :
11         menor_restante;
12 }
```

(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     float matriz[25][25];  
3     float menores[25];  
4     char nome_arquivo[100];  
5     printf("Digite o nome do arquivo:\n");  
6     scanf("%s", nome_arquivo);  
7     preencher_matriz(matriz, nome_arquivo);  
8     for (int j = 0; j < 25; j++) {  
9         menores[j] = encontrar_menor_valor(matriz,  
10             j, 0, 25);  
11    }  
12    printf("Matriz:\n");  
13    for (int i = 0; i < 25; i++) {  
14        for (int j = 0; j < 25; j++) {  
15            printf("%.2f ", matriz[i][j]);  
16        }  
17        printf("\n");  
18    }  
19    printf("Vetor dos menores valores de cada  
20        coluna:\n");  
21    for (int j = 0; j < 25; j++) {  
22        printf("%.2f ", menores[j]);  
23    }  
24}
```