

```

1 // Bibliotecas Utilizadas
2 #include <stdio.h> // Para entrada e saída padrões
3 #include <stdlib.h> // Biblioteca padrão
4
5 // Dados de Produção
6 /*
7     Clientes:
8         Universidade Estadual do Maranhão
9         Centro de Ciências Tecnológicas
10        Departamento de Engenharia da Computação
11        Curso de Engenharia da Computação
12        Disciplina: Estrutura de Dados Básica (ASL092N321)
13        Semestre: 2022.2          Turma: 01
14        Professores:
15            Luís Carlos Fonseca
16            Pedro Brandão Neto
17
18        Autores:
19            Alunos:
20                Alexsandro Lucena Mota    Código: 20210024710
21
22        Propósito do Programa:
23
24        Implementar uma função miniMax(v,n,a,b) que receba um vetor v,
25        contendo n números, e devolva a e b, respectivamente, os valores
26        mínimo e máximo entre aqueles armazenados em v.
27
28        Dados de Manutenção do programa
29
30        Data          Programador          Descrição da Mudança
31        =====
32        2022/4/9      Alexsandro Lucena Mota - Código original (versão 0.1).
33        2022/5/9      Alexsandro Lucena Mota - Acrescimento de comentário para
34                                           facilitar a manutenção do programa.
35        2022/5/9      Alexsandro Lucena Mota - Adaptando o programa para alocação
36                                           dinâmica.
37        2022/5/9      Alexsandro Lucena Mota - Implementado o teste de alocação de memória;
38                                           Implementado a liberação de memória;
39                                           Implementado, após liberação de memória alocada,
40                                           a atribuição NULL aos ponteiros para evitar que
41                                           fiquem soltos.
42 */
43
44 // Lista de Protótipos de Funções e Procedimentos
45 int miniMax();
46
47 // Escopo da Função Main
48 int main(){
49     // Apresentação do programa ao usuário
50     printf("PROGRAMA miniMax\n\n");
51     printf("Propósito:\n");
52     printf("\tRecebe um vetor de dimensão n e retorna o me-\n");
53     printf("\tnor e o maior valor entre suas componentes.\n\n");
54
55     // Entrada de dados
56     printf("Entre com a dimensão do vetor!\n");
57     int dim = 0; // inicializa a variável dim com valor 0.
58     printf("DIM = ");
59     scanf("%d", &dim); // usuário informa a dimensão do vetor (array).
60     printf("Entre com as componentes (valores inteiros) do vetor!\n");
61     int *vetor; // ponteiro vetor (aponta para um endereço de memória.
62     vetor = (int *) malloc (dim*sizeof(int)); /* - alocando memória para vetor;
63                                           - o vetor é do tipo inteiro
64                                           - sizeof(int) retorna o quantidade
65                                           de bytes ocupados pelo tipo int: 4bytes.
66                                           - dim dar o número de componentes do vetor;

```

```

67                                     dim multiplica o sizeof(int) provendo o
68                                     número de bytes necessário para alocar o
69                                     vetor
70                                     */
71     if(vetor == NULL){ // teste de alocação de memória
72         printf("Erro: Memória Insuficiente!\n");
73         exit(1);
74     }
75     for(int i = 0; i < dim; i++){
76         scanf("%d", vetor + i); // usuário informa os valores de entrada do vetor
77     }
78
79     // Ecoando as informações entrada
80     printf("As componentes informadas foram:\n");
81     for(int i = 0; i < dim; i++){
82         printf("v[%d] = %d\n", i + 1, *(vetor + i));
83     }
84
85     // Saída de dados
86     int *return_vector; // para passagem de parâmetros de referência na função miniMax
87     return_vector = (int *) malloc (2*sizeof(int)); // retornará somente duas componentes
88     if(return_vector == NULL){ // teste de alocação de memória
89         printf("Erro: Memória Insuficiente!\n");
90         exit(1);
91     }
92     miniMax(vetor,dim,return_vector);
93     free(vetor); // liberando a memória alocada
94     vetor = NULL; // para evitar que o ponteiro fique solto
95     printf("O menor valor entre as componentes é: %d.\n", *return_vector);
96     printf("O maior valor entre as componentes é: %d.\n", *(return_vector + 1));
97     free(return_vector); // liberando a memória alocada
98     return_vector = NULL; // para evitar que o ponteiro fique solto
99
100    // Retorno da função
101    return 0;
102 }
103
104 // Escopo das funções e dos procedimentos
105 int miniMax(int*in_vector,int num, int*out_vector){
106     /*
107         Função miniMax
108
109         Propósito:
110             Recebe um vetor, sua dimensão, e retornar o menor
111             e maior valor (inteiros) entre suas componentes.
112     */
113
114     // Atribuições necessárias
115     int minnum = *in_vector; // incializa a variável local minnum
116     int maxnum = *in_vector; // incializa a variável local maxnum
117
118     // Rotina para determinar o menor e o maior valor
119     for(int i = 1; i < num; i++){
120         if( *(in_vector + i) > maxnum){
121             maxnum = *(in_vector + i);
122         }else if( *(in_vector + i) < minnum){
123             minnum = *(in_vector + i);
124         }
125     }
126
127     /*
128         Atribuido as saídas aos respectivos endereços de memória
129         para acesso externo a função miniMax
130     */
131     *(out_vector + 0) = minnum;
132     *(out_vector + 1) = maxnum;

```

```
133
134     // Retorno da função
135     return 0;
136 }
```