

```

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 */
38
39 // Lista de Protótipos de Funções Procedimetnos
40 int miniMax();
41
42 // Escopo da Função Main
43 int main(){
44     // Apresentação do programa ao usuário
45     printf("PROGRAMA miniMax\n\n");
46     printf("Propósito:\n");
47     printf("\tRecebe um vetor de dimensão n e retorna o me-\n");
48     printf("\tnor e o maior valor entre suas componentes.\n\n");
49
50     // Entrada de dados
51     printf("Entre com a dimensão do vetor!\n");
52     int dim = 0; // incializa a variável dim com valor 0.
53     printf("DIM = ");
54     scanf("%d", &dim); // usuário informa a dimensão do vetor (array).
55     printf("Entre com as componentes (valores inteiros) do vetor!\n");
56     int *vetor; // ponteiro vetor (aponta para um endereço de memória.
57     vetor = (int *) malloc (dim*sizeof(int)); /* - alocando memória para vetor;
58                                          - o vetor é do tipo inteiro
59                                          - sizeof(int) retorna o quantidade
60                                          de bytes ocupados pelo tipo int: 4bytes.
61                                          - dim dar o número de componentes do vetor;
62                                          dim multiplica o sizeof(int) provendo o
63                                          número de bytes necessário para alocar o
64                                          vetor
65                                          */
66     for(int i = 0; i < dim; i++){

```

```

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

```