

[illegible]

```

67                                     de bytes ocupados pelo tipo int:
68 4bytes.
69                                     - dim dar o número de componentes
70                                     dim multiplica o sizeof(int)
71                                     número de bytes necessário para
72                                     vetor
73                                     */
74 if(vetor == NULL){ // teste de alocação de memória
75     printf("Erro: Memória Insuficiente!\n");
76     exit(1);
77 }
78 for(int i = 0; i < dim; i++){
79     scanf("%d", vetor + i); // usuário informa os valores de entrada do vetor
80 }
81 // Ecoando as informações entrada
82 printf("As componentes informadas foram:\n");
83 for(int i = 0; i < dim; i++){
84     printf("v[%d] = %d\n", i + 1, *(vetor + i));
85 }
86
87 // Saída de dados
88 int *return_vector; // para passagem de parâmetros de referência na função
miniMax
89 return_vector = (int *) malloc (2*sizeof(int)); // retornará somente duas
componentes
90 if(return_vector == NULL){ // teste de alocação de memória
91     printf("Erro: Memória Insuficiente!\n");
92     exit(1);
93 }
94 miniMax(vetor, dim, return_vector);
95 free(vetor); // liberando a memória alocada
96 vetor = NULL; // para evitar que o ponteiro fique solto
97 printf("O menor valor entre as componentes é: %d.\n", *return_vector);
98 printf("O maior valor entre as componentes é: %d.\n", *(return_vector + 1));
99 free(return_vector); // liberando a memória alocada
100 return_vector = NULL; // para evitar que o ponteiro fique solto
101
102 // Retorno da função
103 return 0;
104 }
105
106 // Escopo das funções e dos procedimentos
107 int miniMax(int*in_vector, int num, int*out_vector){
108     /*
109     Função miniMax
110
111     Propósito:
112     Recebe um vetor, sua dimensão, e retornar o menor
113     e maior valor (inteiros) entre suas componentes.
114     */
115
116     // Atribuições necessárias
117     int minnum = *in_vector; // inicializa a variável local minnum
118     int maxnum = *in_vector; // inicializa a variável local maxnum
119
120     // Rotina para determinar o menor e o maior valor
121     for(int i = 1; i < num; i++){
122         if( *(in_vector + i) > maxnum){
123             maxnum = *(in_vector + i);
124         }else if( *(in_vector + i) < minnum){
125             minnum = *(in_vector + i);
126         }

```

```
127     }
128
129     /*
130      Atribuido as saídas aos respectivos endereços de memória
131      para acesso externo a função miniMax
132     */
133     *(out_vector + 0) = minnum;
134     *(out_vector + 1) = maxnum;
135
136     // Retorno da função
137     return 0;
138 }
```