

```

1  /*****
2  Clientes:
3      Universidade Estadual do Maranhão
4      Centro de Ciências Tecnológicas
5      Departamento de Engenharia da Computação
6      Curso de Engenharia da Computação
7      Disciplina: Estrutura de Dados Básica (ASL092N321)
8      Semestre: 2022.2      Turma: 01
9      Professores:
10     Luís Carlos Fonseca
11     Pedro Brandão Neto
12
13 Autores:
14     Alunos:
15     Alexsandro Lucena Mota      Código: 20210024710
16
17 Título: Atividade Prática
18
19 Propósito do Programa:
20
21     Crie uma programa para preencher 2 vetores: Um não ordenado e o outro
22     ordenado. Ambos com tamanho 8. Usar funções. Crie um menu de texto com
23     as seguintes opções:
24         1) Preencher vetor 1
25         2) Preencher vetor 2
26         3) Busca Sequencial (vetor 1)
27         4) Busca Binária (vetor 2)
28         5) Imprimir vetor 1
29         6) Imprimir vetor 2
30         7) Sair.
31
32 Dados de Manutenção do programa
33
34     Data      Programador      Descrição da Mudança
35     =====
36     2022/12/10      Alexsandro Lucena Mota      - Código original (versão 0.1).
37 *****/
38
39 #include <stdio.h>
40 #include <stdlib.h>
41 #define tam 8
42
43 /*****/
44 int menu_select();
45 void preencher_vetor();
46 int busca_sequencial();
47 int busca_binaria();
48 void imprimir_vetor();
49 /*****/
50 int main(){
51     printf("PROGRAMA DOIS VETORES: ORDENADO E NAO ORDENADO\n");
52     int choose = 0, *vetor1 = NULL, *vetor2 = NULL;
53     int elemento_vetor = 0, retorno = 0;
54     vetor1 = (int *) malloc(tam*sizeof(int));
55     if(vetor1 == NULL){ // teste de alocação de memória
56         printf("Erro: Memória Insuficiente!\n");
57         exit(1);
58     }
59     vetor2 = (int *) malloc(tam*sizeof(int));
60     if(vetor2 == NULL){ // teste de alocação de memória
61         printf("Erro: Memória Insuficiente!\n");
62         exit(1);
63     }
64     for(int i = 0; i < tam; i++){
65         *(vetor1 + i) = NULL;
66         *(vetor2 + i) = NULL;

```

```

67     }
68     do{
69         printf("\n");
70         choose = menu_select();
71         switch(choose){
72             case 1:
73                 printf("\n");
74                 printf("Preencha o vetor sem se preocupar com a ordem: ");
75                 preencher_vetor(vetor1);
76                 break;
77             case 2:
78                 printf("\n");
79                 printf("Preencha o vetor de forma ordenada, ou seja, ");
80                 printf("em ordem crescente ou decrescente: ");
81                 preencher_vetor(vetor2);
82                 break;
83             case 3:
84                 printf("Informe o valor que deseja buscar: ");
85                 scanf("%d",&elemento_vetor);
86                 retorno = busca_sequencial(elemento_vetor,vetor1);
87                 if(retorno == -1){
88                     printf("Valor nao encontrado. ");
89                     printf("Nao eh uma componente deste vetor.\n");
90                 }else{
91                     printf("valor encontrado na posicao %d do vetor: ", retorno);
92                     printf("vetor[%d] = %d.\n", retorno, *(vetor1 + retorno));
93                 }
94                 break;
95             case 4:
96                 printf("Informe o valor que deseja buscar: ");
97                 scanf("%d",&elemento_vetor);
98                 retorno = busca_binaria(elemento_vetor,vetor2);
99                 if(retorno == -1){
100                     printf("Valor nao encontrado. ");
101                     printf("Nao eh uma componente deste vetor.\n");
102                 }else{
103                     printf("valor encontrado na posicao %d do vetor: ", retorno);
104                     printf("vetor[%d] = %d.\n", retorno, *(vetor2 + retorno));
105                 }
106                 break;
107             case 5:
108                 printf("\n");
109                 printf("vetor_1 = { ");
110                 imprimir_vetor(vetor1);
111                 printf("}\n");
112                 break;
113             case 6:
114                 printf("\n");
115                 printf("vetor_2 = { ");
116                 imprimir_vetor(vetor2);
117                 printf("}\n");
118                 break;
119             case 7:
120                 printf("\nVoce escolheu sair do programa.\n");
121                 exit(0); //return 0;
122         }
123     }while(choose != 7);
124     free(vetor1);
125     free(vetor2);
126     return 0;
127 }
128 /*****
129 int menu_select(){
130     int ch = 0;
131     do{
132         printf("Menu de Opcoes\n");

```

```

133     printf("\t1 - Preencher vetor 1 (Desordenado)\n");
134     printf("\t2 - Preencher vetor 2 (Ordenado)\n");
135     printf("\t3 - Busca Sequencial (vetor 1)\n");
136     printf("\t4 - Busca Binária (vetor 2)\n");
137     printf("\t5 - Imprimir vetor 1\n");
138     printf("\t6 - Imprimir vetor 2\n");
139     printf("\t7 - Sair.\n");
140     printf("Entre com a opcao desejada: ");
141     scanf("%d",&ch); /** Lê do teclado a seleção */
142     if((ch < 1) || (ch > 7)){
143         printf("\nOpcao Invalida! Tente novamente.\n");
144     }
145     }while( (ch < 1) || (ch > 7) );
146     return ch;
147 }
148 /*-----*/
149 void preencher_vetor(int *vetor){
150     for(int i = 0; i < tam; i++){
151         scanf("%d",vetor + i);
152     }
153 };
154 /*-----*/
155 int busca_sequencial(int valor, int *vetor){
156     printf("\n");
157     for(int i = 0; i < tam; i++){
158         if(valor == *(vetor + i)){
159             return i;
160         }
161     }
162     return -1;
163 };
164 /*-----*/
165 int busca_binaria(int valor,int *vetor){
166     printf("\n");
167     int inicio = 0, fim = tam - 1, pmedia = 0;
168     do{
169         pmedia = (fim + inicio)/2;
170         if(valor == *(vetor + pmedia)){
171             return pmedia;
172         }else if(valor < *(vetor + pmedia)){
173             fim = pmedia - 1;
174         }else{
175             inicio = pmedia + 1;
176         }
177     }while(inicio <= fim);
178     return -1;
179 };
180 /*-----*/
181 void imprimir_vetor(int *vetor){
182     for(int i = 0; i < tam; i++){
183         printf("%d ",*(vetor + i));
184     }
185 };
186 /*-----*/

```