

arvore1-2/main.c

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include "ABP.h"
4
5  int main(){
6
7      ABP* A;
8      int n, matricula;
9      char nome[200];
10     double nota;
11
12     printf("-----\n");
13     printf("ABP Sequencial Dinâmica:\n");
14     printf("1 - Criar ABP;\n");
15     printf("2 - Inserir Aluno;\n");
16     printf("3 - Buscar Aluno;\n");
17     printf("4 - Remover Aluno pelo nome;\n");
18     printf("5 - Imprimir ABP em ordem;\n");
19     printf("6 - Imprimir informações do aluno com maior nota;\n");
20     printf("7 - Imprimir informações do aluno com menor nota;\n");
21     printf("8 - Destruir a ABP;\n");
22     printf("9 - Sair.\n");
23     printf("-----\n");
24
25     do{
26
27         scanf("%d", &n);
28
29         switch(n){
30
31             case 1:
32                 A = criaABP();
33                 if(A != NULL) printf("ABP criada com sucesso.\n");
34                 break;
35
36             case 2:
37                 printf("Matricula do aluno: ");
38                 scanf("%d", &matricula);
39                 printf("Nome do aluno: ");
40                 scanf("%s", nome);
41                 printf("Nota do aluno: ");
42                 scanf("%lf", &nota);
43                 if(inseremmatricula(A, matricula, nome, nota)) printf("Matricula
44 inserido com sucesso.\n");
45                 break;
46
47             case 3:
48                 printf("Nome a ser buscado: ");
49                 scanf("%s", nome);
50                 printf("Matricula do aluno: ");
51                 scanf("%d", &matricula);
52                 if(pesquisa(A, matricula, nome)) printf("Matricula encontrado.\n");
53                 else
54                     printf("Matricula não foi encontrado.\n");
55                 break;
```

```
56         case 4:
57             printf("Nome a ser removido: ");
58             scanf("%s", nome);
59             printf("Matricula do aluno: ");
60             scanf("%d", &matricula);
61             if(removematricula(A, matricula, nome)) printf("Aluno removido com
sucesso.\n");
62             break;
63
64         case 5:
65             em_ordem(*A, 0);
66             break;
67
68         case 6:
69             matricula = achaMaior(*A, (*A)->nota, (*A)->matricula);
70             imprimeMaior(*A, matricula);
71             break;
72
73         case 7:
74             matricula = achaMenor(*A, (*A)->nota, (*A)->matricula);
75             imprimeMenor(*A, matricula);
76             break;
77
78         case 8:
79             destroiABP(A);
80             printf("ABP destruida com sucesso\n");
81             break;
82
83     }
84
85     printf("-----\n");
86
87     } while(n != 9);
88
89     printf("-----\n");
90
91     return 0;
92
93 }
```