

---

```

1  /*
2      Feito por:
3          Felipe Fontenele de Ávila Magalhães
4          Gilmar Marques Serafim Júnior
5          Mateus Vitor Pereira Lana
6          Thiago Oliveira de Santana
7
8      Data início: 21/07/2016      Término: 26/07/2016
9
10     Disciplina: BCC 203 - Estrutura de Dados II
11     Prof.: Guilherme Tavares de Assis
12
13
14 */
15 #include <stdlib.h>
16 #include <stdio.h>
17 #include <string.h>
18 #include <time.h>
19
20 #include "Cadeia.h"
21 #include "ForcaBruta.h"
22 #include "BM.h"
23 #include "BMHS.h"
24 #include "ShiftAnd.h"
25
26 int main(int argc, char* argv[]){
27     if(argc<4 || argc>5){
28         printf("\nFormato de entrada via linha de comando invalido!\n");
29         printf("Formato esperado: <executavel> <metodo> <texto> <padrao> [-P](opcional)\n\n");
30         return 0;
31     }
32
33     clock_t inicio, fim, total;
34     double tempo;
35     int op;
36     char nome[20];
37     long comp=0, desloc=0;
38     op = atoi(argv[1]);
39     strcpy(nome,argv[2]);
40
41     while(op<1 || op>4){
42         printf("\n METODO DE CASAMENTO EXATO DE CADEIAS INVALIDO. FAVOR ESCOLHER UMA OPCAO DO MENU
ABAIXO...\n\n");
43         printf("      =====MENU=====
===== \n");
44         printf("  ||
  ||\n");
45         printf("  || 1 - Forca Bruta.
  ||\n");
46         printf("  || 2 - Boyer-Moore(BM) com a heuristica ocorrencia.
  ||\n");
47         printf("  || 3 - Boyer-Moore-Horspool-Sunday(BHMS).
  ||\n");
48         printf("  || 4 - Shift-And Exato.
  ||\n");
49         printf("  ||
  ||\n");
50         printf("      =====
===== \n\n");
51         printf(" Escolha: ");
52         fflush(stdin);
53         scanf("%d",&op);
54     }
55     while(strcmp(nome, "texto_pequeno") != 0 && strcmp(nome, "texto_medio") != 0 && strcmp(nome, "
texto_grande") != 0){
56         printf("\n NOME DO ARQUIVO INVALIDO. FAVOR DIGITAR UM NOME REFERENTE AS OPCOES ABAIXO...\n\n");

```

---

```

57     printf(" =====MENU=====↵
===== \n");
58     printf(" || ↵
||\n");
59     printf(" || - texto_pequeno ↵
||\n");
60     printf(" || - texto_medio ↵
||\n");
61     printf(" || - texto_grande ↵
||\n");
62     printf(" || ↵
||\n");
63     printf(" =====↵
===== \n\n");
64     printf(" Escolha: ");
65     fflush(stdin);
66     scanf("%s",nome);
67 }
68
69 strcat(nome, ".txt");
70 TipoTexto tipoTexto;
71
72 tipoTexto = lerTexto(nome);
73
74 //padrão a ser pesquisado
75 TipoPadrao tipoPadrao = malloc(sizeof(argv[3]));
76 strcpy(tipoPadrao, argv[3]);
77
78 //avalia qual método vai ser usado
79 switch(op){
80     case 1: { //força bruta
81         inicio = clock();
82         printf("\nMetodo Forca Bruta:\n");
83         if(!ForcaBruta(tipoTexto, strlen(tipoTexto), tipoPadrao, strlen(tipoPadrao), &comp, &desloc)){
84             printf("\nPadrao nao encontrado\n");
85         }
86
87         fim = clock();
88         total = fim - inicio;
89         tempo = ((double)total)/CLOCKS_PER_SEC;
90         if(argv[4] != NULL)
91         {
92             if(strcmp(argv[4], "-P") == 0 || strcmp(argv[4], "-p") == 0)
93             {
94                 printf("\n\n =====RESULTADOS=====↵
=====");
95                 printf("\n || ↵
||");
96                 printf("\n || Metodo de casamento exato de cadeias: Forca Bruta ↵
||");
97                 printf("\n || Comparacoes entre os caracteres do texto e do padrao: %-7ld ↵
||", comp);
98                 printf("\n || Deslocamentos realizados na varredura de todo o texto: %-7ld ↵
||", desloc);
99                 printf("\n || Tempo de execucao do Forca Bruta: %.3f segundos. ↵
||", tempo);
100                 printf("\n || ↵
||");
101                 printf("\n =====↵
===== \n\n");
102             }
103         }
104         break;
105     }
106     case 2: { //boyer-moore
107         inicio = clock();

```

```

108     printf("\nMetodo Boyer-Moore com a heuristica ocorrencia:\n");
109     if(!BM(tipoTexto, strlen(tipoTexto), tipoPadrao, strlen(tipoPadrao), &comp, &desloc)){
110         printf("\nPadrao nao encontrado\n");
111     }
112
113     fim = clock();
114     total = fim - inicio;
115     tempo = ((double)total)/CLOCKS_PER_SEC;
116     if(argv[4] != NULL)
117     {
118         if(strcmp(argv[4], "-P") == 0 || strcmp(argv[4], "-p") == 0)
119         {
120             printf("\n\n      =====RESULTADOS=====↵
=====
");
121             printf("\n      ||↵
||");
122             printf("\n      || Metodo de casamento exato de cadeias: Boyer-Moore↵
||");
123             printf("\n      || Comparacoes entre os caracteres do texto e do padrao: %-7ld↵
||", comp);
124             printf("\n      || Deslocamentos realizados na varredura de todo o texto: %-7ld↵
||", desloc);
125             printf("\n      || Tempo de execucao do Boyer-Moore: %.3f segundos.↵
||", tempo);
126             printf("\n      ||↵
||");
127             printf("\n      =====↵
=====\\n\\n");
128         }
129     }
130     break;
131 }
132 case 3: { //boyer-moore-horspool-sunday
133     inicio = clock();
134     printf("\nMetodo Boyer-Moore-Horspool-Sunday(BMHS):\n");
135     if(!BMHS(tipoTexto, strlen(tipoTexto), tipoPadrao, strlen(tipoPadrao), &comp, &desloc)){
136         printf("\nPadrao nao encontrado\n");
137     }
138
139     fim = clock();
140     total = fim - inicio;
141     tempo = ((double)total)/CLOCKS_PER_SEC;
142     if(argv[4] != NULL)
143     {
144         if(strcmp(argv[4], "-P") == 0 || strcmp(argv[4], "-p") == 0)
145         {
146             printf("\n\n      =====RESULTADOS=====↵
=====
");
147             printf("\n      ||↵
||");
148             printf("\n      || Metodo de casamento exato de cadeias: Boyer-Moore-Horspool-Sunday↵
||");
149             printf("\n      || Comparacoes entre os caracteres do texto e do padrao: %-7ld↵
||", comp);
150             printf("\n      || Deslocamentos realizados na varredura de todo o texto: %-7ld↵
||", desloc);
151             printf("\n      || Tempo de execucao do Boyer-Moore-Horspool-Sunday: %.3f segundos.↵
||", tempo);
152             printf("\n      ||↵
||");
153             printf("\n      =====↵
=====\\n\\n");
154         }
155     }
156     break;
157 }

```

```

158     case 4:{ //shift-and exato
159         inicio = clock();
160         printf("\nMetodo Shift-And Exato:\n");
161         if(!ShiftAnd(tipoTexto, strlen(tipoTexto), tipoPadrao, strlen(tipoPadrao),&comp,&desloc)){
162             printf("\nPadrao nao encontrado\n");
163         }
164
165         fim = clock();
166         total = fim - inicio;
167         tempo = ((double)total)/CLOCKS_PER_SEC;
168         if(argv[4] != NULL)
169         {
170             if(strcmp(argv[4],"[-P]") ==0 || strcmp(argv[4],"[-p]")==0)
171             {
172                 printf("\n\n      =====RESULTADOS=====↵
=====");
173                 printf("\n      ||↵
||");
174                 printf("\n      || Metodo de casamento exato de cadeias: Shift-And Exato↵
||");
175                 printf("\n      || Comparacoes entre os caracteres do texto e do padrao: %-7ld↵
||",comp);
176                 printf("\n      || Deslocamentos realizados na varredura de todo o texto: %-7ld↵
||",desloc);
177                 printf("\n      || Tempo de execucao do Shift-And Exato: %.3f segundos.↵
||",tempo);
178                 printf("\n      ||↵
||");
179                 printf("\n      =====↵
=====\\n\\n");
180             }
181         }
182         break;
183     }
184 }
185
186 free(tipoTexto);
187 free(tipoPadrao);
188 return 0;
189 }
190

```