

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4
5  //?????? ??? ????? BinTreeElementType
6  typedef struct
7  {
8      int code;
9      int recNo;
10 } BinTreeElementType;
11
12 //??????????? ???????? student.
13 typedef struct
14 {
15     int code;
16     char surname[20];
17     char name[20];
18     char sex;
19     int year;
20     float grade;
21 }studentT;
22
23
24 typedef struct BinTreeNode *BinTreePointer;
25 struct BinTreeNode {
26     BinTreeElementType Data;
27     BinTreePointer LChild, RChild;
28 } ;
29
30 typedef enum {
31     FALSE, TRUE
32 } boolean;
33
34
35 void CreateBST(BinTreePointer *Root);
36 boolean BSTEmpty(BinTreePointer Root);
37 void RecBSTInsert(BinTreePointer *Root, BinTreeElementType Item);
38 void RecBSTSearch(BinTreePointer Root, BinTreeElementType KeyValue, boolean *Found, BinTreePointer *LocPtr
);
39 void RecBSTDelete(BinTreePointer *Root, BinTreeElementType KeyValue);
40 void RecBSTInorder(BinTreePointer Root);
41 void RecBSTPreorder(BinTreePointer Root);
42 void RecBSTPostorder(BinTreePointer Root);
43 void menu(int *choise);
44 int BuildBST(BinTreePointer *Root);
45 void InsertNewStudent(BinTreePointer *Root,int *size);
46 void SearchStudent(BinTreePointer Root);
47 void PrintStudent(int RecNum);
48 void PrintStudentsWithGrade();
49
50
51
52 int main()
53 {
54     //?????? ???????????.
55     BinTreePointer ARoot;
56     int size,choise;
57
58     //????????????????? ???????? ??? menou ????? ?? ?????????? ? ???????? ?? ???????????.
59     while(TRUE)
60     {
61         menu(&choise);
62         switch(choise)
63         {
64             //????? ?????????? ??? ? ???????? ?????? 1 ?????? ??? ?????????? ??? ?????????? ???(index).
65             case 1:

```

```

66         size = BuildBST(&ARoot);
67         break;
68         //???? ?????????? ??? ? ??????? ????? 2 ????? ??? ?????????? ??? ?????????? ??? ?????????? ??? ???
??? ??? ????????.
69         case 2:
70             InsertNewStudent(&ARoot,&size);
71             break;
72             //???? ?????????? ??? ? ??????? ????? 3 ????? ??? ?????????? ??? ?????????? ??? ?????????? ???
????????? ??? ?????????? ????.
73         case 3:
74             SearchStudent(ARoot);
75             break;
76             //???? ?????????? ??? ? ??????? ????? 4 ????? ??? ?????????? ??? ?????????? ??? ?????????? ???
?? ????.
77         case 4:
78             printf("\nPrint all students data.\n\n");
79             RecBSTInorder(ARoot);
80             printf("\n");
81             break;
82         case 5:
83             //???? ?????????? ??? ? ??????? ????? 5 ????? ??? ?????????? ??? ?????????? ?????????? ?????????? ??
?????? ?????????????? ??? ??? ??????.
84             PrintStudentsWithGrade();
85             break;
86             //???? ?????????? ??? ? ??????? ????? 6,????? ???????????????.
87         case 6:
88             return 0;
89     }
90 }
91 }
92
93 void CreateBST(BinTreePointer *Root)
94 {
95     *Root = NULL;
96 }
97
98 boolean BSTEmpty(BinTreePointer Root)
99 {
100     return (Root==NULL);
101 }
102
103 void RecBSTInsert(BinTreePointer *Root, BinTreeElementType Item)
104 {
105     if (BSTEmpty(*Root)) {
106         (*Root) = (BinTreePointer)malloc(sizeof (struct BinTreeNode));
107         (*Root) ->Data = Item;
108         (*Root) ->LChild = NULL;
109         (*Root) ->RChild = NULL;
110     }
111     else
112         if (Item.code < (*Root)->Data.code)
113             RecBSTInsert(&(*Root) ->LChild,Item);
114         else if (Item.code > (*Root)->Data.code)
115             RecBSTInsert(&(*Root) ->RChild,Item);
116         else
117             printf("To %d EINAI HDH STO DDA\n", Item.code);
118 }
119
120 void RecBSTSearch(BinTreePointer Root, BinTreeElementType KeyValue,boolean *Found, BinTreePointer *LocPtr)
121 {
122     if (BSTEmpty(Root))
123         *Found=FALSE;
124     else
125         if (KeyValue.code < Root->Data.code)
126             RecBSTSearch(Root->LChild, KeyValue, &(*Found), &(*LocPtr));
127         else

```

```

128         if (KeyValue.code > Root->Data.code)
129             RecBSTSearch(Root->RChild, KeyValue, &(*Found), &(*LocPtr));
130     else
131     {
132         *Found = TRUE;
133         *LocPtr=Root;
134     }
135 }
136
137 void RecBSTDelete(BinTreePointer *Root, BinTreeElementType KeyValue)
138 {
139     BinTreePointer TempPtr;          /* true AN TO STOIXEIO KeyValue EINAI STOIXEIO TOY DDA */
140
141     if (BSEmpty(*Root))              /* ?????? ?????? ?? KeyValue ?? ?? ?????? */
142         printf("to %d DeN BRE8HKe STO DDA\n", KeyValue.code);
143     else
144         /* ?????????? ?????????? ??? ?????? ??? ?????????? ??? ?????? KeyValue ??? ?????????? ???
145         if (KeyValue.code < (*Root)->Data.code)
146             RecBSTDelete(&((*Root)->LChild), KeyValue);          /* ?????????? ?????????? *
147         else
148             if (KeyValue.code > (*Root)->Data.code)
149                 RecBSTDelete(&((*Root)->RChild), KeyValue);      /* ?????? ?????????? *
150             else
151                 /* TO KeyValue ?????????? ?????????? ??? ??????? *)
152                 if ((*Root)->LChild ==NULL)
153                 {
154                     TempPtr = *Root;
155                     *Root = (*Root)->RChild;          /* ??? ?????? ?????????? ?????? *)
156                     free(TempPtr);
157                 }
158                 else if ((*Root)->RChild == NULL)
159                 {
160                     TempPtr = *Root;
161                     *Root = (*Root)->LChild;          /* ?????? ?????????? ??????, ????? ??? ????? *)
162                     free(TempPtr);
163                 }
164                 else
165                     /* ?????? 2 ??????? *)
166                     {
167                         /* ??????? ??? INORDER ?????????? ??? *)
168                         TempPtr = (*Root)->RChild;
169                         while (TempPtr->LChild != NULL)
170                             TempPtr = TempPtr->LChild;
171                         /* ??????????? ??? ?????????? ??? ?????? ??? ???????????
172                         ??? ??????????? ??? ?????????? ??? ?????????? ??????? */
173                         (*Root)->Data = TempPtr->Data;
174                         RecBSTDelete(&((*Root)->RChild), (*Root)->Data);
175                     }
176 }
177
178 void RecBSTInorder(BinTreePointer Root)
179 {
180     if (Root!=NULL) {
181         RecBSTInorder(Root->LChild);
182         printf(" (%d, %d),",Root->Data.code,Root->Data.recNo);
183         RecBSTInorder(Root->RChild);
184     }
185 }
186
187 void RecBSTPreorder(BinTreePointer Root)
188 {
189     if (Root!=NULL) {
190         printf("/%d /",Root->Data.code);
191         printf("L");
192         RecBSTPreorder(Root->LChild);
193         printf("R");

```

```

194     RecBSTPreorder(Root->RChild);
195 }
196 printf("U");
197 }
198
199 void RecBSTPostorder(BinTreePointer Root)
200 {
201     if (Root!=NULL) {
202         printf("L");
203         RecBSTPostorder(Root->LChild);
204         printf("R");
205         RecBSTPostorder(Root->RChild);
206         printf("/%d /",Root->Data.code);
207     }
208     printf("U");
209 }
210
211 /*????????? menu ??? ?????????? ??? ?????????? ??? ???? ? ?????????? ??? ??????????
212 ??? ?????????? ??? ???? ??? ????????????? ?????? choise.?? ? ???? ??? ?? ?????? ???
213 ?????? ??????? 1-6 ,????????? ?????????? ???????.*/
214 void menu(int *choise)
215 {
216     printf("\n                MENOY                \n");
217     printf("-----\n");
218     printf("1. Build BST\n");
219     printf("2. Insert new student\n");
220     printf("3. Search for a student\n");
221     printf("4. Print all students (Traverse Inorder)\n");
222     printf("5. Print students with a >= given grade\n");
223     printf("6. Quit\n");
224
225     do{
226         printf("\nEpilogh: ");
227         scanf("%d",&(*choise));
228
229         if(*choise < 1 || *choise > 6)
230             printf("Number must be between 1-6.Try again.\n");
231     }while(*choise < 1 || *choise > 6);
232 }
233
234 int BuildBST(BinTreePointer *Root)
235 {
236     //?????? ??????????
237     FILE *infile;
238     studentT student;
239     BinTreeElementType indexRec;
240     int size,nscan;
241
242     //????????????? ???.
243     CreateBST(&(*Root));
244
245     //????????? ??? ?????????? ??? ??????????.
246     infile = fopen("foitites.dat","r");
247
248     //????????????????? ??? ?????????? ??????????
249     size = 0;
250
251     //?? ?????????? ?????? ?? ???????.
252     if(infile != NULL)
253         while(TRUE)
254         {
255             //????????? ??? ?????????? ???????-??????? ??? ?????????? ??? ?????????? ??? ?????????? ??? ??????????
256             nscan = fscanf(infile,"%d, %20[^,], %20[^,], %c, %d, %f\n",&student.code, student.surname,
257                 student.name, &student.sex, &student.year, &student.grade);
258             if(nscan == EOF) break;

```

```

258         if(nscan != 6)
259         {
260             printf("Improper file format.\n");
261             break;
262         }
263         //??????? ??? ?? ??? ??? ?????? ?????? ??? ?????????? ? ???????? ??? ??????????(index).
264         indexRec.code = student.code;
265         indexRec.recNo = size;
266
267         //????????? ??? ????.
268         RecBSTInsert(&(*Root),indexRec);
269
270         //??????? ??? ?????? ??? 1.
271         size++;
272     }
273     //????????? ??? ??????????.
274     fclose(infile);
275
276     //??????????? ??? ??????????.
277     return size;
278 }
279
280 void InsertNewStudent(BinTreePointer *Root,int *size)
281 {
282     //??????? ????????????.
283     FILE *infile;
284     studentT student;
285     BinTreeElementType indexRec;
286     boolean found;
287     BinTreePointer LocPtr;
288
289     //????????? ??? ?????????? ??? ????????????.
290     infile = fopen("foitites.dat","a");
291
292     //??????????? ??? ?? ??? ??? ????????????.
293     printf("Give students AM: ");
294     scanf("%d",&indexRec.code);
295
296     //??????????? ??? ????.
297     RecBSTSearch(*Root,indexRec,&found,&LocPtr);
298
299     //?? ??? ??????????
300     if(!found)
301     {
302         //????????? ?? ?????????? ??? ??????????.
303         student.code = indexRec.code;
304
305         printf("Give student surname: ");
306         scanf("%s",student.surname);
307
308         printf("Give student name: ");
309         scanf("%s",student.name);
310
311         printf("Give student's registration year: ");
312         scanf("%d",&student.year);
313
314         printf("Give student's grade: ");
315         scanf("%f",&student.grade);
316
317         printf("Give student sex F/M: ");
318         scanf(" %c",&student.sex);
319
320         indexRec.recNo = *size;
321
322         //??????????? ??? ????.
323         RecBSTInsert(&(*Root),indexRec);

```

```

324
325         //???????? ??? ??????.
326         fprintf(infile,"%d, %s, %s, %c, %d, %.1f\n", student.code, student.surname, student.name,
student.sex, student.year, student.grade);
327
328         //?????? ??? ?????? ??? 1.
329         size++;
330     }
331     //?? ??? ??????? ??????? ??????? ????????.
332     else
333         printf("Afto to AM yparxei hdh.\n");
334
335     //???????? ??? ??????.
336     fclose(infile);
337 }
338
339 void SearchStudent(BinTreePointer Root)
340 {
341     //?????? ??????????.
342     BinTreeElementType indexRec;
343     BinTreePointer LocPtr;
344     boolean found;
345
346     //???????? ??? ?? ??? ??????????.
347     printf("\nGive student's code: ");
348     scanf("%d",&indexRec.code);
349     printf("\n");
350
351     //???????? ??? ????.
352     RecBSTSearch(Root,indexRec,&found,&LocPtr);
353
354     //?? ??????? ????????? ????????? ?? ??? ??????? ??? ?????????? PrintStudent.
355     if(found)
356         PrintStudent(LocPtr->Data.recNo);
357     //?? ??? ??????? ????????? ????????? ??????????.
358     else
359         printf("Student NOT found.\n");
360 }
361
362 void PrintStudent(int RecNum)
363 {
364     //?????? ??????????.
365     FILE *infile;
366     int nscan,lines;
367     studentT student;
368
369     //???????? ??? ??????? ??? ??????????.
370     infile = fopen("foitites.dat","r");
371
372     //?????????????? ??? ??????????? ??? ?????? ??? ??????? ??? ?????????????
373     lines = 0;
374
375     //?? ??????? ?????? ?? ??????.
376     if(infile != NULL)
377     {
378
379         while(lines <= RecNum)
380         {
381             //???????? ??? ??????? ??????-?????? ??? ??????? ??? ?????????? ??? ??????? ?????? ??? ?????????
Student.
382             nscan = fscanf(infile,"%d, %20[^,], %20[^,], %c, %d, %f\n",&student.code, student.surname,
student.name, &student.sex, &student.year, &student.grade);
383             if(nscan == EOF) break;
384             if(nscan != 6)
385             {
386                 printf("Improper file format.\n");

```

```

387         break;
388     }
389     //?????? ??? ?????? ????.
390     lines++;
391 }
392 //????? ?????? ??? ?????? ??? ?????? ?????? ??? ?????? ?????? ??? ??????
??? ??????.
393     if(lines)
394 printf("%d,%s,%s,%c,%d,%.1f\n",student.code,student.name,student.surname,student.sex,student.year,student.grade)
;
395     }
396     //????????? ??? ??????.
397     fclose(infile);
398 }
399 void PrintStudentsWithGrade()
400 {
401     //?????? ??????????.
402     FILE *infile;
403     int nscan;
404     studentT student;
405     float theGrade;
406
407     //????????? ??? ?????? ?? ??? ?????? ?????? ?? ??????????.
408     printf("Give the grade: ");
409     scanf("%f",&theGrade);
410
411     //????????? ??? ?????? ??? ??????.
412     infile = fopen("foitites.dat","r");
413
414     //????????????? ??????.
415     printf("Student's with grade >= %.1f\n\n",theGrade);
416
417     //?? ?????? ?????? ?? ??????.
418     if(infile != NULL)
419         while(TRUE)
420         {
421             //????????? ??? ?????? ??????-????? ??? ?????? ??? ?????? ??? ?????? ??? ??????
Student.
422             nscan = fscanf(infile,"%d,%20[^,], %20[^,], %c, %d, %f\n",&student.code, student.surname,
student.name, &student.sex, &student.year, &student.grade);
423             if(nscan == EOF) break;
424             if(nscan != 6)
425             {
426                 printf("Improper file format.\n");
427                 break;
428             }
429             //?? ? ?????? ??? ?????????? ????? >= ?????? ??? ?????? ??? ?????? ?? ?????????? ?? ?????????
???.
430             if(student.grade >= theGrade)
431 printf("%d,%s,%s,%c,%d,%.1f\n",student.code,student.name,student.surname,student.sex,student.year,student.grade)
;
432         }
433         printf("\n");
434
435         //????????? ??? ??????.
436         fclose(infile);
437     }
438

```