

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  #define megisto_plithos 26
5
6  typedef enum {
7      FALSE, TRUE
8  } boolean;
9
10 typedef boolean typos_synolou[megisto_plithos];
11 typedef int stoixeio_synolou;
12
13 void Katholiko(typos_synolou synolo);
14 void EmfanisiSynolou(typos_synolou set);
15 boolean Melos(stoixeio_synolou stoixeio, typos_synolou synolo);
16 void Dimiourgia(typos_synolou synolo);
17 void Eisagogi(stoixeio_synolou stoixeio, typos_synolou synolo);
18 boolean IsaSynola(typos_synolou s1, typos_synolou s2);
19 boolean Yposynolo(typos_synolou s1, typos_synolou s2);
20 void EnosiSynolou(typos_synolou s1, typos_synolou s2, typos_synolou enosi);
21 void TomiSynolou(typos_synolou s1, typos_synolou s2, typos_synolou tomi);
22 void DiaforaSynolou(typos_synolou s1, typos_synolou s2, typos_synolou diafora);
23
24 int main()
25 {
26     //?????? ???????
27     typos_synolou Kath,S,T,Union,Intersection,Difference;
28
29     /*????????????? ?????????????? ??????? Kath ?? ?? ??????????
30     ?????????? ??? ?????????? ?????????? ??? ? ?????????? ???
31     (???????? a)
32     */
33     Katholiko(Kath);
34     printf("Synolo Kefalaiwn Agglikwn Grammatwn = ");
35     EmfanisiSynolou(Kath);
36
37     /*????????????? ?????? ????????? S,????????? ??? ?????????? A B
38     C D ??? ?????????? ???.(???????? b)*/
39     Dimiourgia(S);
40     printf("S=");
41     for(stoixeio_synolou i=0;i < 4; i++)
42         Eisagogi(i,S);
43     EmfanisiSynolou(S);
44
45     /*????????????? ?????? ????????? T,????????? ??? ?????????? A C
46     E G I ??? ?????????? ???.(???????? b)*/
47     Dimiourgia(T);
48     printf("T=");
49     for(stoixeio_synolou i=0; i < 9; i+=2)
50         Eisagogi(i,T);
51     EmfanisiSynolou(T);
52
53     /*????????? ?? ?? ??????? S ??? ? ?????? ??? ? ?????????? ??
54     ??? ?? ?? ????.(???????? c)
55     */
56     if(IsaSynola(S,T))
57         printf("ISA SYNOLA.\n");
58     if(Yposynolo(S,T))
59         printf("S YPOSYNOLO T.\n");
60     if(Yposynolo(T,S))
61         printf("T YPOSYNOLO S.\n");
62
63     /*????? ??? ?? ?????????? ?????? ??????? ?????????? ??? ?????? ???
64     ????????? S ??? T ??? ??????? Union ??? ?? ??????????????.(???????? d)*/
65     EnosiSynolou(S,T,Union);
66     printf("ENWSI SYNOLWN S KAI T =");

```

```

67     EmfanisiSynolou(Union);
68
69     /*???? ???? ?? ?????????? ???? ?????? ????????? ???? ???? ???
70     ??????? S ??? T ??? ?????? Intersection ??? ?? ?????????????.(??????? e)*/
71     TomiSynolou(S,T,Intersection);
72     printf("TOMH SYNOLWN S KAI T =");
73     EmfanisiSynolou(Intersection);
74
75     /*???? ???? ?? ?????????? ???? ?????????? ????????? ???? ????????? ????
76     ??????? S-T ??? ?????? Difference ??? ?? ?????????????.(??????? f)*/
77     DiaforaSynolou(S,T,Difference);
78     printf("DIAFORA SYNOLWN S-T=");
79     EmfanisiSynolou(Difference);
80
81     return 0;
82 }
83
84 void Katholiko(typos_synolou synolo)
85 {
86     stoixeio_synolou i;
87
88     for (i = 0; i < megisto_plithos; i++)
89         synolo[i] = TRUE;
90 }
91 /*????????? ???? ???? ?????????????? ???? ?????? ?? ?? ????????? ???? ????
92 ??? ?????????????? ???? ????????? ???? ?????????? ???? ??????????*/
93 void EmfanisiSynolou(typos_synolou set)
94 {
95     stoixeio_synolou i;
96     char alphabet[]="ABCDEFGHIJKLMNOPQRSTUVWXYZ";
97
98     printf("{");
99     for (i=0;i < megisto_plithos;i++)
100         if(Melos(i,set))
101             printf("%c ",alphabet[i]);
102
103     printf("}\n");
104 }
105
106 boolean Melos(stoixeio_synolou stoixeio, typos_synolou synolo)
107 {
108     return synolo[stoixeio];
109 }
110
111 void Dimiourgia(typos_synolou synolo)
112 {
113     stoixeio_synolou i;
114
115     for (i = 0; i < megisto_plithos; i++)
116         synolo[i] = FALSE;
117 }
118
119 void Eisagogi(stoixeio_synolou stoixeio, typos_synolou synolo)
120 {
121     synolo[stoixeio] = TRUE;
122 }
123
124 boolean IsaSynola(typos_synolou s1, typos_synolou s2)
125 {
126     stoixeio_synolou i;
127     boolean isa;
128
129     isa = TRUE;
130     i=0;
131     while (i < megisto_plithos && isa)
132         if (Melos(i,s1) != Melos(i,s2))

```

```

133         isa = FALSE;
134     else
135         i++;
136     return isa;
137 }
138
139 boolean Yposynolo(typos_synolou s1, typos_synolou s2)
140 {
141     stoixeio_synolou i;
142     boolean yposyn;
143
144     yposyn = TRUE;
145     i=0;
146     while (i < megisto_plithos && yposyn)
147         if (Melos(i, s1) && !Melos(i, s2))
148             yposyn = FALSE;
149         else
150             i++;
151     return yposyn;
152 }
153
154 void EnosiSynolou(typos_synolou s1, typos_synolou s2, typos_synolou enosi)
155 {
156     stoixeio_synolou i;
157
158     for (i = 0; i < megisto_plithos; i++)
159         enosi[i] = Melos(i, s1) || Melos(i, s2);
160 }
161
162 void TomiSynolou(typos_synolou s1, typos_synolou s2, typos_synolou tomi)
163 {
164     stoixeio_synolou i;
165
166     for (i = 0; i < megisto_plithos; i++)
167         tomi[i] = Melos(i, s1) && Melos(i, s2);
168 }
169
170 void DiaforaSynolou(typos_synolou s1, typos_synolou s2, typos_synolou diafora)
171 {
172     stoixeio_synolou i;
173
174     for (i = 0; i < megisto_plithos; i++)
175         diafora[i] = Melos(i, s1) && (!Melos(i, s2));
176 }

```