

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4
5  #define megisto_plithos 256
6
7  typedef enum {
8      FALSE, TRUE
9  } boolean;
10
11 typedef boolean typos_synolou[megisto_plithos];
12 typedef int stoixeio_synolou;
13
14 void Dimiourgia(typos_synolou synolo);
15 void Katholiko(typos_synolou synolo);
16 void Eisagogi(stoixeio_synolou stoixeio, typos_synolou synolo);
17 void Diafrafi(stoixeio_synolou stoixeio, typos_synolou synolo);
18 boolean Melos(stoixeio_synolou stoixeio, typos_synolou synolo);
19 boolean KenoSynolo(typos_synolou synolo);
20 boolean IsaSynola(typos_synolou s1, typos_synolou s2);
21 boolean Yposynolo(typos_synolou s1, typos_synolou s2);
22 void EnosiSynolou(typos_synolou s1, typos_synolou s2, typos_synolou enosi);
23 void TomiSynolou(typos_synolou s1, typos_synolou s2, typos_synolou tomi);
24 void DiaforaSynolou(typos_synolou s1, typos_synolou s2, typos_synolou diafora);
25 boolean isValidInteger (char str[]);
26 boolean isValidIdentifier (char str[]);
27
28 int main()
29 {
30     typos_synolou CharacterSet, DigitSet, LetterSet;
31     char answer = 'y', str[30];
32
33
34     Dimiourgia(CharacterSet);
35     Dimiourgia(DigitSet);
36     Dimiourgia(LetterSet);
37
38     Eisagogi(43, CharacterSet);
39     Eisagogi(45, CharacterSet);
40
41     for(int i=48; i < 58; i++)
42         Eisagogi(i, DigitSet);
43
44     for(int i=65; i < 91; i++)
45         Eisagogi(i, LetterSet);
46
47     for(int i=97; i < 123; i++)
48         Eisagogi(i, LetterSet);
49
50     while(answer == 'y')
51     {
52         printf("Give an integer to check if it is valid:");
53         scanf(" %s", &str);
54
55         if(isValidInteger(str))
56             printf("Is Valid\n");
57         else
58             printf("Not Valid\n");
59
60         printf("Do you want more y/n:");
61         scanf(" %c", &answer);
62
63     }
64
65
66     /*??? ???? ????? ??? ??? ?? ?????*/

```

```

67
68
69
70 }
71
72 void Dimiourgia(typos_synolou synolo)
73
74 {
75     stoixeio_synolou i;
76
77     for (i = 0; i < megisto_plithos; i++)
78         synolo[i] = FALSE;
79 }
80
81 void Katholiko(typos_synolou synolo)
82 {
83     stoixeio_synolou i;
84
85     for (i = 0; i < megisto_plithos; i++)
86         synolo[i] = TRUE;
87 }
88
89 void Eisagogi(stoixeio_synolou stoixeio, typos_synolou synolo)
90 {
91     synolo[stoixeio] = TRUE;
92 }
93
94 void Diagrafi(stoixeio_synolou stoixeio, typos_synolou synolo)
95 {
96     synolo[stoixeio] = FALSE;
97 }
98
99 boolean Melos(stoixeio_synolou stoixeio, typos_synolou synolo)
100 {
101     return synolo[stoixeio];
102 }
103
104 boolean KenoSynolo(typos_synolou synolo)
105 {
106     stoixeio_synolou i;
107     boolean keno;
108
109     keno=TRUE;
110     i = 0;
111
112     while (i < megisto_plithos && keno) {
113         if (Melos(i, synolo))
114             keno = FALSE;
115         else
116             i++;
117     }
118     return keno;
119 }
120
121 boolean IsaSynola(typos_synolou s1, typos_synolou s2)
122 {
123     stoixeio_synolou i;
124     boolean isa;
125
126     isa = TRUE;
127     i=0;
128     while (i < megisto_plithos && isa)
129         if (Melos(i,s1) != Melos(i,s2))
130             isa = FALSE;
131         else
132             i++;

```

```

133     return isa;
134 }
135
136 boolean Yposynolo(typos_synolou s1, typos_synolou s2)
137 {
138     stoixeio_synolou i;
139     boolean yposyn;
140
141     yposyn = TRUE;
142     i=0;
143     while (i < megisto_plithos && yposyn)
144         if (Melos(i, s1) && !Melos(i, s2))
145             yposyn = FALSE;
146         else
147             i++;
148     return yposyn;
149 }
150
151 void EnosiSynolou(typos_synolou s1, typos_synolou s2, typos_synolou enosi)
152 {
153     stoixeio_synolou i;
154
155     for (i = 0; i < megisto_plithos; i++)
156         enosi[i] = Melos(i, s1) || Melos(i, s2);
157 }
158
159 void TomiSynolou(typos_synolou s1, typos_synolou s2, typos_synolou tomi)
160 {
161     stoixeio_synolou i;
162
163     for (i = 0; i < megisto_plithos; i++)
164         tomi[i] = Melos(i, s1) && Melos(i, s2);
165 }
166
167 void DiaforaSynolou(typos_synolou s1, typos_synolou s2, typos_synolou diafora)
168 {
169     stoixeio_synolou i;
170
171     for (i = 0; i < megisto_plithos; i++)
172         diafora[i] = Melos(i, s1) && (!Melos(i, s2));
173 }
174
175 boolean isValidInteger (char str[])
176 {
177     boolean valid = TRUE;
178
179     if(str[0] == 43 || str[0] == 45)
180         for (int i=1; str[i] != '\0'; i++)
181             if(str[i] >= 48 && str[i] < 58)
182                 valid = TRUE;
183             else
184                 {
185                     valid = FALSE;
186                     break;
187                 }
188
189     else
190         valid = FALSE;
191
192     return valid;
193 }
194 }
195
196 boolean isValidIdentifier (char str[])
197 {
198

```

```
199
200
201 /*?? ??? ????? ????? ??? ??? ????? ,???? ??? if ??? ??? for ?? ?????????
202 (>= 65 && < 91) || (>= 97 && < 123)*/
203
204
205 }
```