

Contents

1	count.c	2
2	shift.c	4

1 count.c

```
1  /*
2   * count.c
3   *
4   * Created on: Nov 5, 2014
5   * Author: mutazmanaa
6   */
7
8
9  #include <stdio.h>
10
11  #define SPACE ' '
12  #define NEW_LINE '\n'
13
14  /**
15   main function : program that count leeters & words & lines.
16   */
17  int main()
18  {
19      int charsCounter = 0, charsCounterTemp = 0, wordsCounter = 0, linesCounter = 1;
20      char letter = 0;
21
22      while (scanf("%c", &letter) != EOF)
23      {
24
25
26          if (letter == SPACE || letter == NEW_LINE) //counting spaces & \n as chars and the temp get zero
27          {
28              charsCounterTemp = 0;
29              charsCounter++;
30
31          }else
32          {
33
34              charsCounterTemp++; //when letter not space that is a begining og the word
35              charsCounter++;
36
37          }
38
39
40
41
42
43          if (charsCounterTemp == 1) //when begin with letter that is a world
44          {
45              wordsCounter++;
46          }
47
48          if (letter == NEW_LINE) // new line with \n without space at end of line.
49          {
50
51              linesCounter++;
52          }
53
54
55
56
57
58      }
59  }
```

```
60
61
62     printf("%d %d %d", charsCounter, wordsCounter, linesCounter);
63
64     return 0;
65 }
```

2 shift.c

```
1  /*
2   * shift.c
3   *
4   * Created on: Nov 4, 2014
5   * Author: mutazmanaa
6   */
7
8
9  #include <stdio.h>
10
11
12  #define ENCRYPTION 'e'
13  #define DECREPTION 'd'
14  #define MAX_UPPER_CASE 'Z'
15  #define MIN_UPPER_CASE 'A'
16  #define MAX_LOWER_CASE 'z'
17  #define MIN_LOWER_CASE 'a'
18  #define MIN_NUMBER '0'
19  #define MAX_NUMBER '9'
20  #define NEW_LINE '\n'
21  #define N 25
22  #define MODULO 10
23  #define MINUS -1
24
25  char decrypt (int shift, char c);
26  char encrypt (int shift, char c);
27  char shiftCalc(int shift,char c, char left, char right);
28
29  /**
30   * function that dealing with encryptions.
31   * @param shift a movement from the original letter/character
32   * @param c is a char be decrypted
33   * @return a char after be decrypted
34   */
35
36  char encrypt (int shift, char c)
37  {
38      char result = 0;
39
40
41      if(c >= MIN_NUMBER && c <= MAX_NUMBER)
42      {
43
44          result = shiftCalc(shift%MODULO, c, MIN_NUMBER, MAX_NUMBER);
45      }
46      else if(c >= MIN_LOWER_CASE && c<= MAX_LOWER_CASE)
47      {
48
49          result = shiftCalc(shift,c,MIN_LOWER_CASE, MAX_LOWER_CASE);
50
51      }
52      else if(c>= MIN_UPPER_CASE && c<= MAX_UPPER_CASE)
53      {
54
55          result = shiftCalc(shift, c, MIN_UPPER_CASE, MAX_UPPER_CASE);
56      }
57      else
58      {
59
```

```

60         result = c;
61     }
62
63     return result;
64 }
65
66
67
68 /**
69  * function that dealing with deryptions.
70  *@param shift a movement from the original letter/character
71  *@param c is a char be encrypted
72  *@return a char after be encrypted
73  */
74
75
76 char decrypt(int shift, char c)
77 {
78
79     return encrypt(MINUS*shift, c); // revers function of decrypt or decrypt with negative shif
80 }
81
82 /**.
83  * function that caculate the right char after decryption and deal with dependencies.
84  *@param shift a movement from the original letter/character
85  *@param c is a char be encrypted
86  *@param left is the start limit numbers/letters
87  *@param right is the end limit numbers/letters
88  */
89 char shiftCalc(int shift, char c, char left, char right)
90 {
91
92     int remain = 0;
93     if (shift > N || shift < MINUS*N)
94     {
95         return 0;
96     }
97     int res = ((int)c) + shift;
98
99     if (res > right) // deal with dependencies.
100     {
101         remain = res - right;
102         res = left + remain - 1;
103     }
104
105     if(res < left)
106     {
107         remain = left - res;
108         res = right - remain +1;
109     }
110
111     return (char)res;
112
113 }
114
115
116
117 int main()
118 {
119     int shift = 0;
120     char selection = 0;
121     char c = 0;
122
123
124     printf("please enter the shift:\n");
125
126     scanf("%d", &shift);
127

```

```

128
129     do {
130         printf("would you like to encrypt (e) or decrypt (d)?\n");
131         scanf(" %c", &selection);
132
133     } while(selection != ENCRYPTION && selection != DECREPTION);
134
135
136     //printf("\n");
137
138     while (scanf(" %c", &c) != EOF)
139     {
140
141
142
143         if(c == NEW_LINE)
144         {
145             continue;
146         }
147
148         if(selection == DECREPTION)
149         {
150             printf("%c", decrypt(shift, c));
151         }
152         else
153         {
154             printf("%c", encrypt(shift, c));
155         }
156
157     }
158
159     printf("\n");
160
161
162
163
164
165     return 0;
166
167 }

```