LAB 12/26

1. Write two subroutines void rfill1D(…) and void rfill2D(…) to meet following requirements.
   a) The size of 1D data array is K=10, and rfill1D will randomly fill the given 1D array with length K using non-repeating integers from 0 to (K-1).
   b) The size of 2D data array is 5-by-4, and rfill2D will randomly fill the given 5-by-4 2D array using non-repeating integers from 0 to (5*4-1).
   c) Write a subroutine void print1DArray(…), which prints out the given 1D array.
   d) Write a subroutine void print2DArray(…), which prints out the given 2D array.
   e) Write a subroutine void sort1D(…), which sorts the given 1D array in descending order.

   Refer to following code to complete your main program to verify the subroutines.

```
int main() {

    // Something missing here…

    printf("\nrfill1D:\n");
    rfill1D( ... );
    print1DArray( ... );


    printf("\nAfter sorting:\n");
    sort( ... );
    print1DArray( ... );
    printf("\n");



    printf("\nrfill2D:\n");
    rfill2D( ... );
    print2DArray( ... );

    return 0;
}
```

Your output should look like this: (the result will be different, for sure)

```
rfill1D:
 3  8  0  4  6  7  9  5  1  2
After sorting:
 9  8  7  6  5  4  3  2  1  0

rfill2D:
 5  4  2  6
13 10  9 17
 8 11 15 19
18  3 16  7
12 14  0  1
```

2. IIUPC 2009
Problem D: Digital Fortress

In the last IIUPC there was a problem called Da Vinci Code prepared on the story of the bestselling book of Dan Brown, The Da Vinci Code. Here is another problem based on his another techno-thriller novel Digital Fortress. In this problem, you will be given a cipher text. Your task is to decipher the text using the decrypting technique described below. Let's take an example. A cipher text is given as follows:

WECGEWHYAAIORTNU

The output will be:

WEAREWATCHINGYOU

For this problem, there are 16 characters in the given cipher text "WECGEWHYAAIORTNU" which is square of 4. These letters have to be arranged in nxn (in this example 4 x 4) grid and each letter from the given input will be placed in a grid in row major order (1st row, 2nd row, 3rd row, …). When the given cipher text is placed in the grid it looks like as follow:

```
W  E  C  G
E  W  H  Y
A  A  I  O
R  T  N  U
```

From the above grid if we take the letters in column major order (1st column, 2nd column, 3rd column, …) then we get the following decrypted text:

WEAREWATCHINGYOU

Input

Input starts with a line consisting of a single number T. T test cases follow. Each test case consists of one line. This line contains the cipher text. The cipher text contains either UPPERCASE letters or blank spaces. Total number of character in the text will not be more 10,000.

Output

For each test case, the output contains a single line containing the decrypted text. If the number of characters in the input text is not square of any number, then give the output "INVALID".

Sample Input

3
WECGEWHYAAIORTNU
DAVINCICODE
DTFRIAOEGLRSI TS

Output for Sample Input

WEAREWATCHINGYOU
INVALID
DIGITAL FORTRESS

Please write a function void decode(…) that has two parameters accepting char array with cipher text and length of this cipher text, and then prints out the decrypted text.

3. Please solve maze problem as following requirement.
a) Implement the function **showmap(…)** which accepts one 2D array with map information and show map on screen. Mark start point and end point as below figure. The map data and size are in the file, **Lab11_20171226_3_maze.h**.

b) Implement the recursive function **trace(…)** which accepts three parameters, one 2D array with map information, one tracing position x, and one tracing position y, traces and printout the path of maze from start position to end position.

```c
#include <stdio.h>
#include <conio.h>
#include <conio2.h>
#include <stdlib.h>
// You need to include maze.h

int Sy=1, Sx=1;   // Start Position
int Ey=23, Ex=78; // End Position
int success=0;    // Flag for record if find the end position sucessfully
int sp=0, ry[1000], rx[1000]; // Record Path Position



int main(void)
{
  // call showmap
  getch();

  // call trace
  if (success == 0)
    printf("No Path!!!\n");

  getch();
  return 0;
}

// Complete this function
...... showmap(......)
{
  .........
}


// Complete this function
... trace(...)
{
  map[y][x]=1;  // mark as traced path
  ry[sp]=y; rx[sp]=x; sp++;   // Store tracing Path

  //if reaching the End position,
  //show tracing path in rx and ry, and let success flag be 1
```

```
  if (y==Ey && x==Ex){
      for (int k=0; k<sp; k++) {
          gotoxy(rx[k]+1,ry[k]+1);
          textbackground(LIGHTRED);

          if (ry[k]==Sy && rx[k]==Sx) {
              textbackground(LIGHTBLUE);
              printf("S");
              continue;
          }
          if (ry[k]==Ey && rx[k]==Ex) {
              textbackground(LIGHTGREEN);
              printf("E");
              continue;
          }
          printf(" ");
          _sleep(150);
      }
      success = 1;
  }

  // if not reach the end position(by judging success flag)
  // and if the wanted direction is not path, trace the direction by recursion
  // There are four direction to trace, up, right, down, left
  ...Complete code here...

  sp--;
  map[y][x]=0;
}
```