PROGRAMMING AND PROBLEM SOLVING (SE 1105) FINAL	**A**	Grading				
		Q1	Q2	Q3	Q4	Σ

Instructors	ID #	Name-Surname	Time allowed	Date/Room #
Dr. Dindar ÖZ Dr. Faegheh YEGANLI		ANSWER KEY	80 mins.	December 29, 2022 (10:40-12:00)

Notes: If you believe that necessary data or assumptions are missing from the problem statement, make your own assumption(s) and write them clearly.

QUESTIONS

1. **(30 pts.)** Write the outputs of the following programs.

a) (15 pts)

```c
#include <stdio.h>

void funcA(char **pptr, int *ptr)
{
   *pptr += *ptr;
}

void main()
{
   int arr[] = { 6,0,2,3,4,3,1};
   char str[]= "Happy new year!";
   char *left= str;

   for (int i=0; *left!=0; i++)
   {
     printf("%s\n",left);
     funcA(&left, (arr + i));
   }
}
```

b) (15 pts)

```c
#include <stdio.h>

int funcB(int *arr, int n)
{
   if (arr[0]==-1)
       return n;
   printf("[%d]\n",*arr);
   return funcB(arr+arr[1], n+arr[0]);
}

void main()
{
   int arr[]=
      {2,2,0,4,-1,8,2,3,1,3,3,1,-1};

   int res= funcB(arr,0)
   for (int i = 0; i < res; ++i) {
     printf("=");
   }
}
```
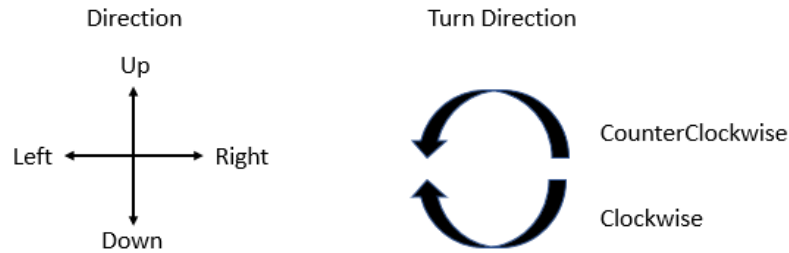
Happy new year!
new year!
new year!
w year!
ear!

[2]
[0]
[2]
[3]
=======

2. **(20 pts.)**
Define a macro DIRECTIONCOUNT as 4
Define two enumerated types Direction and TurnDirection as described in the pictures below.

Direction

Up

Left ←——→ Right

Down

Turn Direction

CounterClockwise

Clockwise

Write a function that takes a Direction (*initialDirection*), an array of TurnDirection (turns) and the size of the array (*turnCount*) as parameters. The function should return the final direction after making the given turns, assuming that the amount of turn for each one is 90 degrees.(i.e Left becomes Up after one clockwise turn.)

Example: If initialDirection: Up   turns = { Clockwise, Clockwise, CounterClockwise, Clockwise, ClockWise }
then it returns Left

```c
#define DIRECTIONCOUNT 4
typedef enum {Up, Right, Down, Left} Direction;
typedef enum {Clockwise=1, CounterClockwise=3} TurnDirection;

Direction q2( Direction initialDirection, TurnDirection turns[],
            int turnCount)
{
    int totalTurn=0;
    for (int i = 0; i < turnCount; ++i) {
        totalTurn += turns[i];
    }
    return (initialDirection+totalTurn)%DIRECTIONCOUNT;
}
```

PROGRAMMING AND PROBLEM SOLVING (SE 1105) FINAL	A	Grading				
		Q1	Q2	Q3	Q4	Σ

Instructors	ID #	Name-Surname	Time allowed	Date/Room #
Dr. Dindar ÖZ Dr. Faegheh YEGANLI		ANSWER KEY	80 mins.	December 29, 2022 (10:40-12:00)

3. **(25 pts.)** A two dimensional integer array is used as a map to represent an island in an ocean, where 0 denotes water of the ocean and 1 denotes the land of the island.There is no lake in the island all lands are connected (i.e. there is a single island).The map is surrounded by water. Define a new data type Map as 10x10 2D integer array. Write a function that takes a Map and returns the circumference of the island on the map.(see the example.) Take the length of the edge of one cell of the map as 1.

Example:
If the map were 4x4 as follows: map : [[0,1,1,0], [0,0,1,0],[0,0,1,0],[0,1,1,1,]] output: 16 (The number of bold edges on the figure is the circumference)

0	1	1	0
0	0	1	0
0	0	1	0
0	1	1	1

.

```c
typedef int Map[10][10];

int circumf(Map map, int r, int c)
{
    int circum =0;
    if (r<=0 || map[r-1][c]==0)
        circum++;
    if (c<=0 || map[r][c-1]==0)
        circum++;
    if (r>=9 || map[r+1][c]==0)
        circum++;
    if (c>=9 || map[r][c+1]==0)
        circum++;
    return circum;
}
```

```c
int q4(Map map)
{
  int crc = 0;
  for (int r = 0; r < 10; r++)
  {
    for (int c = 0; c < 10; c++) {
      if (map[r][c]==1)
        crc += circumf(map, r, c);
    }
  }
  return crc;
}
```

PROGRAMMING AND PROBLEM SOLVING (SE 1105) FINAL	A	Grading				
		Q1	Q2	Q3	Q4	Σ

Instructors	ID #	Name-Surname	Time allowed	Date/Room #
Dr. Dindar ÖZ Dr. Faegheh YEGANLI		ANSWER KEY	80 mins.	December 29, 2022 (10:40-12:00)

4. **(25 pts)** Write a function that takes a string (*str*) which contains only small English letters and a positive integer n as parameters. The function must return if there exists any letter which occurs in str n times or more.

Example: If str: "thisisaneasyquestion" n: 3 then returns **true** ('i' takes place 3 times, 's' takes place 4 times)
If str: "notsoeasy" n:3 then returns **false** (no letter takes place more than twice)

```
// SOLUTION-1 (with pointers)
bool q4(char * str,int n)
{
    int letterCounts[26]= {0};// Frequency of each letter

    while(*str!=NULL) {
        letterCounts[*str - 'a']++;
        if (letterCounts[*str-'a']>=n)
            return true;
        str++;
    }
    return false;
}




// SOLUTION-2 (without pointers)
bool q4_b(char str[],int n)
{
    for (char c = 'a'; c <='z' ; c++) {
        for (int count=0,i = 0; str[i]!='\0'; ++i) {
            if (str[i]==c)
            {
                if (++count>=n)
                    return true;
            }
        }
    }
    return false;
}
```

Good luck...