PROGRAMMING AND PROBLEM SOLVING (SE 1105) FINAL	A	Grading				
		Q1	Q2	Q3	Q4	Σ
Instructors	**ID #**	**Name-Surname**		**Time**	**Date/Room #**	
Dr. Dindar ÖZ Dr. Kazım ERDOĞDU Dr. Mete EMİNAĞAOĞLU				90 mins.	January 16, 2025	

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. **(25 pts.)** The function below returns the season, n day after the given month in Türkiye. Define the necessary types, macros, global variables (if any) used in the function.

```
#define YEARSEASONS 4
#define SEASONMONTHS 3
#define YEARMONTHS 12
#define MONTHDAYS 30

enum Month {Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec}
typedef enum Month Month;

enum {Summer, Fall, Winter, Spring} Season;
typedef enum Season Season;

Month FirstMonths[] = {Jun,Sep,Dec,Mar};
```

```
/**
 *   Months of the year are:
 *   Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec
 *   Seasons of the year are : Summer, Fall, Winter, Spring
 *   Starting months of the seasons : Jun, Sep, Dec, Mar
*/
Season seasonAfter(Month m, int n)
{
    // Finding the month n days after the given month
    // Assuming each month takes 30 days and each year takes 12 months

    Month  nextMonth = (m + (n / MONTHDAYS)) % YEARMONTHS;

    // Finding the difference between next month and
    // the starting month of summer in terms of month

    int delta = nextMonth - FirstMonths[Summer];
    if (delta<0) delta += YEARMONTHS;

    // Finding the season delta months after the next month
    // Assuming each seasons takes 3 months and each year takes 4 seasons

    Season  season = (Summer + (delta/SEASONMONTHS)) % YEARSEASONS;
    return season;
}
```

PROGRAMMING AND PROBLEM SOLVING (SE 1105) FINAL	A	Grading				
		Q1	Q2	Q3	Q4	Σ
Instructors	**ID #**	**Name-Surname**		**Time**	**Date/Room #**	
Dr. Dindar ÖZ Dr. Kazım ERDOĞDU Dr. Mete EMİNAĞAOĞLU		ANSWER KEY		90 mins.	January 16, 2025	

2. **(25 pts.)** Write a C function that takes a **string (null-terminated char array)** as a parameter and returns **true** if the string contains balanced '(' and ')', or false otherwise. If every '(' has a corresponding ')' in the correct order, the string contains balanced '(' and ')'.

Examples:

```
Input: a                          Output: true
Input: (a + b)                    Output: true
Input: (a + (b*c) – (d + e))      Output: true
Input: (a + b))                   Output: false
Input: (a + b) ) *(c / (d-e)      Output: false
Input: )a + b(                    Output: false
```

ANSWER:

```
bool balancedParanthesis(char s[])
{
    int count = 0;

    for(int i=0; s[i] != '\0' && count >= 0; i++)
    {
        if (s[i] == '(')
            count++;
        else if (s[i] == ')')
            count--;

    }

    return (count == 0);
}
```

PROGRAMMING AND PROBLEM SOLVING (SE 1105) FINAL	A	Grading				
		Q1	Q2	Q3	Q4	Σ
Instructors	**ID #**	**Name-Surname**		**Time**	**Date/Room #**	
Dr. Dindar ÖZ Dr. Kazım ERDOĞDU Dr. Mete EMİNAĞAOĞLU				90 mins.	January 16, 2025	

3. **(25 pts.)** Write a C function that **finds** each and every **row-wise or column-wise pattern** within a given two-dimensional 10 x10 array as the parameter of the function and **returns the total number of such patterns**. The pattern contains **contiguous three identical digits**. *For instance, for the 7 x 7 array given below, the patterns (that are denoted in bold in gray background) must be found and the total number of patterns would be 6.*

7	**7**	**7**	**7**	9	7	0
6	5	0	1	4	4	0
2	5	1	**5**	**5**	**5**	6
1	8	6	**5**	3	4	9
8	3	5	**5**	2	1	3
8	2	2	9	6	1	6
8	0	2	5	**9**	**9**	**9**

Answer:

```
#define row 10
#define col 10
int findPattern(int a[row][col])
{
   int temp = 0;
   int count = 0;
   // Horizontal Patterns
   for (int i=0; i<= row-1; i++)
   {
     for (int j=0; j<= col-3; j++)
     {
       temp = a[i][j];
       if (a[i][j+1] == temp && a[i][j+2] == temp) count++;
     }
   }
  // Vertical Patterns
   for (int j=0; j<= col-1; j++)
   {
     for (int i=0; i<= row-3; i++)
     {
       temp = a[i][j];
       if (a[i+1][j] == temp && a[i+2][j] == temp) count++;
     }
   }
   return count;
}
```

PROGRAMMING AND PROBLEM SOLVING (SE 1105) FINAL	A	Grading				
		Q1	Q2	Q3	Q4	Σ
Instructors	**ID #**	**Name-Surname**		**Time**	**Date/Room #**	
Dr. Dindar ÖZ Dr. Kazım ERDOĞDU Dr. Mete EMİNAĞAOĞLU				90 mins.	January 16, 2025	

4. (25 pts.) Write the outputs of the following programs

a) (10pts)

```
#include<stdio.h>
void beatIt(int *t, int *x)
{
    int m;
    m = *t;
    *t = *x;
    *x = m;
}

int main()
{
    int a[] = {30, 50, 10, 40};
    int done = 0;
    int i;
    while (done == 0)
    {
        done  = 1;
        for (i = 0; i <= 2; i++)
        {
            if (a[i] < a[i+1])
            {
                beatIt(&a[i], &a[i+1]);
            }
        }
        for (i = 3; i >= 1; i--)
        {
            if (a[i] > a[i-1])
            {
                beatIt(&a[i], &a[i-1]);
            }
        }
    }
    printf("%d %d", *(a+1), *(a+3));
}
```

Answer:
40 10

b) (15pts)

```
#include <stdio.h>
int Total = 0;

void coolOne (int a, int b)
{
        int tmp;
        tmp = a;
        a = b;
        b = tmp;
        Total = a + b;
 }
void coolTwo (int*a, int*b)
{
        int tmp;
        tmp = *a;
        *a = *b;
        *b = tmp;
}
void whopper(int n1, int n2)
{
    if (n1 >= n2)
       coolOne (n1 + 1, n2);
    if (n1 < n2)
       coolTwo (&n1, &n2);
    printf ("%d %d %d \n", n1, n2, Total);
}

int main ()
{
       whopper(5,7);
       whopper(-2,-3);
}
```

Answer:
7 5 0
-2 -3 -4

Good luck…