PROGRAMMING AND PROBLEM SOLVING (SE 1105) MIDTERM	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.	October 26, 2022 (18:40-20:00) Y-011, Y107, Y111

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) (15pts)

```
#include "stdio.h"

void main()
{
    int nums[] = {0,1,2,3,4,5};
    int sums[6];
    sums[0]= nums[0];

    printf("{");

    for(int i=1;i<6;i++)
    {
        if (i%2==0)
            sums[i] = sums[i-1]+nums[i];
        else
            sums[i] = sums[i-1]-nums[i];

        printf("%d,",sums[i]);
    }

    printf("}");
}
```

```
{-1,1,-2,2,-3,}
```

b) (15pts)

```
#include "stdio.h"

int func2(int a,int b)
{
    printf("%d %d\n",a+4,b-5);
    a = a+b;
    b = b-1;
    return a+b;
}

int func1(int a[], int b)
{
    printf("%d %d %d\n",a[b], b, a[a[0]]);
    b=b+1;
    a[1]= func2(a[b],b);
    printf("%d %d %d\n",a[b], b, a[a[0]]);
    return a[1]+b;
}

int main() {
    int a[]= {1,2,3,4,5};
    a[3]=func1(a,0);
    for(int i=1; i<5;i+=2)
        printf("%d-",a[i]);
    return 0;
}
```

```
1 0 2
6 -4
3 1 3
3-4-
```

PROGRAMMING AND PROBLEM SOLVING (SE 1105) MIDTERM	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.	October 26, 2022 (18:40-20:00) Y-011, Y107, Y111

2. **(20 pts.)** Write a function that takes an integer parameter n and calculates and returns the following:

$$\big|\,|2 \times n - 7| - |3 \times n - 5|\,\big|$$

(| x | means the absolute value of x)

Note: You can not call any functions unless you implement that function here yourself.

```
int abs(int n)
{
    if (n<0)
        n=-n;
    return n;
}

int q2(int n)
{
    return abs(abs(2*n-7) - abs(3*n-5));
}
```

PROGRAMMING AND PROBLEM SOLVING (SE 1105) MIDTERM	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.	October 26, 2022 (18:40-20:00) Y-011, Y107, Y111

3. **(25 pts.) The neighbor sum** of an element in an array is the sum of the elements that are adjacent to that element (i.e., the Neighbor sum of a[i] = a[i-1]+ a[i+1] if those indices exist). Write a function that takes an array of real numbers (`double`) and its size as parameters. The function should return the element that has the maximum neighbor sum.
(**Example-1:** If the array= `{ 1.0, 9.1, -2.1, 3.5, 7.8, 8.0, -3.1, 10.0 }` Then the element `-3.1` has the maximum neighbor sum: `8.0 + 10.0` so function returns `-3.1`)
(**Example-2:** If the array= `{ 0.5, 6.3, -2.1, -3.5, 1.8, -8.0, 2.1 }` Then the element `0.5` has the maximum neighbor sum: `6.3` so function returns `0.5`)

Note: You can assume that the size of the array will be at least 2, and all elements have different neighbor sums

```
double q3(double  arr[], int len)
{
    int maxIndex=0; // first element
    double max= arr[1];

    if (arr[len-2]>max) // Last element
    {
        maxIndex= len-1;
        max = arr[len-2];
    }
    // Elements in between
    for (int i=1; i<len-1;i++)
    {
        double neighborSum = arr[i-1]+ arr[i+1];
        if (neighborSum>max)
        {
            max= neighborSum;
            maxIndex=i;
        }
    }
    return arr[maxIndex];
}
```

PROGRAMMING AND PROBLEM SOLVING (SE 1105) MIDTERM	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.	October 26, 2022 (18:40-20:00) Y-011, Y107, Y111

4. **(25 pts**.) A rabbit jumps ***m*** times on the first day of the year. After that, each day, it makes ***n*** jumps more than the previous day. If the rabbit ever makes more than 50 jumps in one day, it gets tired and rests for the next day (makes 0 jumps) and continues jumping the same way after the next day.
Write a function that takes m and n as parameters and prints the number of jumps the rabbit is making on each day line by line until the day that the rabbit makes its 1000th jump of the year. (including that day)

```
void q4(int m, int n)
{
    int total = 0;

    for (int jump=m; total<1000; )
    {
        printf("%d\n",jump);
        total+=jump;
        if (jump>50)
            jump=0;
        else jump += n;
    }
}
```

Good luck…