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

Instructors	ID #	Name-Surname	Time	Date/Room #
Dr. Dindar ÖZ Dr. Kazım ERDOĞDU Dr. Mete EMİNAĞAOĞLU			80 mins.	November 19, 2024 (09:40-..)

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 C programs.

a) (15pts)

```
#include <stdio.h>

int g(int m, int M[], int n)
{
    int i = 0;
    while (i < n)
    {
        if (m == M[i])
        {
            return i;
        }
        i++;
    }
    return -1;
}

void f(int A[], int B[], int n)
{
    int i = 0, p = A[0];
    while (B[i] != A[0])
    {
        i = g(B[i], A, n);
        printf("%d --> %d\n", p, A[i]);
        p = A[i];
    }
    printf("%d --> %d\n", p, B[i]);
}

void main()
{
    int A[] = {6, 0, 3, 1, 5, 2, 4};
    int B[] = {1, 2, 4, 5, 3, 0, 6};
    f(A, B, 7);
}
```

Answer:

```
6 --> 1
1 --> 5
5 --> 3
3 --> 4
4 --> 6
```

b) (15pts)

```
#include <stdio.h>

void main()
{
    int i = 1, j, par1, par2;
    int lastr = 5;

    par1 = lastr + 1;

    while (i <= lastr)
    {
        par2 = par1;
        while (par2 >= 1)
        {
            printf(" ");
            par2--;
        }

        j = 1;
        while (j <= i)
        {
            printf("%d ", i);
            j++;
        }

        printf("\n");
        par1--;
        i++;
    }
}
```

Answer:

```
    1
   2 2
  3 3 3
 4 4 4 4
5 5 5 5 5
```

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

Instructors	ID #	Name-Surname	Time	Date/Room #
Dr. Dindar ÖZ Dr. Kazım ERDOĞDU Dr. Mete EMİNAĞAOĞLU			80 mins.	November 19, 2024 (09:40-..)

2. **(20 pts.)** Write a C function that receives a number of seconds as an integer and prints its equivalent hours, minutes, and seconds information on the screen. If the value of hours, minutes, or seconds is 0 then this information should not be printed on the screen (except the very special case of 0 seconds). Check the following examples. Assume that the integers passed to the function are non-negative so you do not need to check whether the given input is negative.

Examples:

Input: 12345 **Output:** 3 h 25 m 45 s
Input: 37 **Output:** 37 s
Input: 120 **Output:** 2 m
Input: 18000 **Output:** 5 h
Input: 7218 **Output:** 2 h 18 s
Input: 10820 **Output:** 3 h 17 m
Input: 620 **Output:** 10 m 20 s.
Input: 0 **Output:** 0

```
void convertSeconds(int seconds)
{
     if (seconds == 0 )
     {
         printf("0");
         return;
     }
     int minutes = seconds / 60;
     seconds = seconds % 60;
     int hours = minutes / 60;
     minutes = minutes % 60;
     if (hours > 0)
     {
          printf("%d h(s) ",hours);
     }
     if (minutes > 0)
     {
          printf("%d m(s) ",minutes);
     }
     if (seconds > 0)
     {
          printf("%d s(s)",seconds);
     }
}
```

PROGRAMMING AND PROBLEM SOLVING (SE 1105) MIDTERM	A	Grading				
		Q1	Q2	Q3	Q4	Σ
Instructors	**ID #**	**Name-Surname**		**Time**	**Date/Room #**	
Dr. Dindar ÖZ Dr. Kazım ERDOĞDU Dr. Mete EMİNAĞAOĞLU				80 mins.	November 19, 2024 (09:40-..)	

3. **(25 pts.)** Write a C function that reads two positive integers (a and b) from the user and prints all perfect squares between a and b. Assume that the user always enters positive integers and a smaller number first. (Hint: Perfect squares are 1,4,9,16,25…)

Attention: You cannot use <math.h> and sqrt() function in your solution.

Example:

```
Enter the first number:   2
Enter the second number: 18
Perfect squares between 2 and 18 are: 4 9 16

Enter the first number:   5
Enter the second number: 90
Perfect squares between 5 and 90 are: 9 16 25 36 49 64 81
```

```
void printPerfects()
{
    int num1, num2;
    printf("Enter the first integer:\n");
    scanf("%d", &num1);
    printf("Enter the second integer:\n");
    scanf("%d", &num2);

    bool started = false;
    int i = 1;
    while (i * i <= num2)
    {
        if (started || i * i >= num1)
        {
            started = true;
            printf("%d\n", i * i);
        }
        i++;
    }
}
```

PROGRAMMING AND PROBLEM SOLVING (SE 1105) MIDTERM	A	Grading				
		Q1	Q2	Q3	Q4	Σ
Instructors	**ID #**	**Name-Surname**		**Time**	**Date/Room #**	
Dr. Dindar ÖZ Dr. Kazım ERDOĞDU Dr. Mete EMİNAĞAOĞLU				80 mins.	November 19, 2024 (09:40-..)	

4. **(25 pts**.) Write a C function that takes the measured average daily temperature of a city for a full month (30 days) as an integer array of length 30. The function returns the biggest temperature change after the hottest day of the month that occurs between two consecutive days. Assume that there is only one hottest day in the array

Example: If the array: { 10, 15, 12, 14, 13, 12 } then it must return 15-12 = 3
If the array: { 10, 12, 18, 14, 11, 10, 15, 12 } then it must return 15-10=5
If the array: { 12, 11, 12, 13, 13, 14, 10, 15 } then it must return 0 as there is no temp change after 15.

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


int biggest(int temps[])
{
    int hottest = 0, biggest = 0;
    int i = 1;
    int delta = 0;
    while (i < 30)
    {
        delta = abs(temps[i] - temps[i - 1]);
        if (delta > biggest) biggest = delta;
        if (temps[i] > temps[hottest])
        {
            hottest = i;
            biggest = 0;
        }
        i++;
    }
    return biggest;
}
```

Good luck…