IMPORTANT NOTES:
1) Only the sentences written in English will be graded. Thus, do not write your answers in any other language.
2) This is a 85 minutes closed books/notes exam.

QUESTIONS

Question-1: Write down the output of the following program.

(a) (10p)
```c
#include <stdio.h>
#define SIZE 20
int main()
{
    int i, j=0;
    int arr[SIZE] = {1, 2, 2, 2, 33, 25, 21, 45, 16, 16,
                     43, 48, 55, 91, 15, 22, 72, 4, 19, 20};
    int arr_new[SIZE];
    for(i=0; i<SIZE; i++)
    {
        if(arr[i]==arr[i+1])
            continue;
        arr_new[j]= arr[i];
        j++;
    }
    for(i=0; i< j; i++)
        printf("%3d", arr_new[i]);
    return 0;
}
```

1  2 33 25 21 45 16 43 48 55 91 15 22 72 4 19 20

(b) (10p)
```c
#include <stdio.h>
void func1(int);
void func2(int, int);
int x=7, y=7;
int main()
{
    int x=4;
    printf("x: %d\n", x);
    printf("y: %d\n", y);
    printf("x: %d\n", x);
    printf("y: %d\n", y);
    printf("x: %d\n", x);
    printf("y: %d\n", y);
    func1(x);
    y=func2(x,y);
    printf("x: %d\n", x);
    printf("y: %d\n", y);
    return 0;
}
```
(c) (10p)
#include <stdio.h>
int main()
{
    int i, j, x, y=7;
x = (y+1)/2;
for(i=1; i<=y; i++) {
    if(i==1 || i==x || i==y) {
        for(j=1; j<=x; j++)
            printf("* ");
    }
    printf("\n");
}
return 0;}

Question-2: Write the program code for the following equation to calculate and display the result. (30p)
y = \sum_{i=1}^{10} \sum_{j=1}^{5} \left( ij - \prod_{k=1}^{7} jk \right)
Question-3: Write a function named `greatest_prime_factor`, which takes an integer as input, and returns the greatest prime number that divides the input integer without remainders. (Greatest prime factor examples: 7 is the greatest prime factor of 28, 2 is the greatest prime factor for 16, 13 is the greatest prime factor of 26, 3 is the greatest prime factor of 12, etc.)

```c
int greatest_prime_factor(int num)
{
    int k, l, prime, gpf;
    for ( k = 1 ; k <= num ; k++){
        if ( num % k == 0 ){ /* Control for division */
            prime = 1; /* Assume k is a prime number */
            for ( l = 2 ; l < k ; l++ ) { /* Check all numbers larger than 1 and smaller than k */
                if ( k % l == 0 ) /* If k is divided by any of them, */
                    prime = 0; /* then k is not a prime number */
            }
            if (prime) /* If k divides num, and k is a prime number*/
                gpf = k; /* Then, the new gpf is k */
        }
    }
    return gpf; /* Return the last (largest) gpf */
}
```
Question-4: Write a program code which takes a size 20 array as input from the user. The program then should consider this array as a vector and calculate its Euclidian distance to the origin, i.e. to a size 20 array of zeros. (Hint: In three dimensions, the Euclidian distance of a vector \{ x, y, z \} to the origin is \( \sqrt{x^2 + y^2 + z^2} \), you should do a similar operation in 20 dimensions, using a loop)

```c
#include <stdio.h>
#include <math.h>
#define SIZE 20
int main()
{
    int i, myarray[SIZE];
    double sum = 0, euc;
    /* Fill the array with inputs from the user */
    for (i = 0; i < SIZE; i++) {
        printf("Enter a number for the array element myarray[%d]\n", i);
        scanf("%d", &myarray[i]);
    }
    /* Print the array (Not necessary) */
    printf("Printing the array:\n");
    for (i = 0; i < SIZE; i++) {
        printf("%3d" , myarray[i]);
    }
    /* Calculate Euclidian distance */
    for (i = 0; i < SIZE; i++) {
        sum += myarray[i] * myarray[i];
    }
    euc = sqrt(sum);
    /* Print Euclidian distance */
    printf("Euclidian distance of this array to the origin is: ");
    printf("%f", euc);
    return 0;
}
```

Good luck.
Yrd. Doç. Dr. Aysun TAŞYAPI ÇELEBİ
Yrd. Doç. Dr. Alp ERTÜRK