```
/*****************************************************************************/
/*HW04_121044074_Onur_Sezer                                                  */
/*_________________________                                                  */
/*Written by Onur Sezer on October 14, 2014                                  */
/*                                                                           */
/*Description:                                                               */
/*____________                                                               */
/*Program kullanicidan alinan degerlerle bir serinin toplamini hesaplama,    */
/*  ucgen cizme, agac cizme, sifreleme, en buyuk ucuncu karekteri bulma      */
/*  islemlerini yapar.
*/
/*Inputs:                                                                    */
/*Konsoltan alinan degerler, enc.txt ve dec.txt                              */
/*                                                                           */
/*Outputs:                                                                   */
/*Konsola basar                                                              */
/*****************************************************************************/

/*---------------------------------------------------------------------------*/
/*                            Includes                                       */
/*---------------------------------------------------------------------------*/
#include<stdio.h>
#include<math.h>
/*---------------------------------------------------------------------------*/
/*                            #defines                                       */
/*---------------------------------------------------------------------------*/
#define MIN 97       /* 'a' nin ASCII degeri */
#define MAX 122      /* 'z' nin ASCII degeri */
#define MIN2 65      /* 'A' nin ASCII degeri */
#define MAX2 90      /* 'Z' nin ASCII degeri */
/*********************************************************/
/*Fonksiyon belirli bir serinin toplamini hesaplar     */
/*input: int m                                          */
/*output:Sonucu return yapar                            */
/*********************************************************/
double compute_serie(int m);
/*********************************************************/
/*Fonksiyon ucgen cizer                                 */
/*input: int size_triangle                              */
/*output:Konsola basar                                  */
/*********************************************************/
void draw_triangle(int size_triangle);
/*********************************************************/
/*Fonksiyon girilen sayi kadar ucgen cizer              */
/*input: int size_triangle,count_triangle               */
/*output:Konsola basar                                  */
/*********************************************************/
void draw_multi_triangle(int size_triangle,int count_triangle);
/*********************************************************/
/*Fonksiyon agacin kokunu cizer                         */
/*input: int size_root                                  */
/*output:Konsola basar                                  */
/*********************************************************/
void draw_root(int size_root);
/*********************************************************/
/*Fonksiyon alinan degerlere gore agac cizer            */
/*input: int size_triangle,count_triangle,size_root     */
/*output:Konsola basar                                  */
/*********************************************************/
void draw_tree( int size_triangle,int count_triangle,int size_root);
/*********************************************************/
/*Fonksiyon sifreleme yapar                             */
/*input: char filename[]                                */
/*         int n                                        */
/*output:Konsola basar                                  */
/*********************************************************/
void encrypt(const char filename[], int n);
/*********************************************************/
/*Fonksiyon sifre cozme yapar                           */
/*input: char filename[]                                */
/*         int n                                        */
/*output:Konsola basar                                  */
```

```c
/*#########################################################*/
void decrypt(const char filename[], int n);
/*#########################################################################*/
/*Fonksiyon girilen karekterlerden en buyuk ucuncu olanini bulur */
/*input: Konsoldan alinir                                                  */
/*output:Konsola basar                                                     */
/*#########################################################################*/
void third_largest();

int main() {
    /*START_OF_MAIN*/
    int m;
    int size_triangle;
    int count_triangle;
    int size_root;
    double result;
    /*END_OF_VARIABLES*/

    /*//////////////////////////////////////*/
    printf("input for compute_serie:\n");
    scanf ("%d",&m );
    result = compute_serie(m);
    printf("Result is: %f\n",result);
    puts("------------------------------");
    /*//////////////////////////////////////*/


    /*//////////////////////////////////////*/
    printf("input for draw_triangle:\n");
    scanf ("%d", &size_triangle );
    draw_triangle(size_triangle);
    puts("------------------------------");
    /*//////////////////////////////////////*/


    /*//////////////////////////////////////*/
    printf("input for draw_multi_triangle:\n");
    scanf ("%d%d",&size_triangle,&count_triangle );
    draw_multi_triangle(size_triangle,count_triangle);
    puts("------------------------------");
    /*//////////////////////////////////////*/


    /*//////////////////////////////////////*/
    printf("input for draw_tree:\n");
    scanf ("%d%d%d",&size_triangle,&count_triangle,&size_root);
    draw_tree(size_triangle, count_triangle, size_root);
    puts("------------------------------");
    /*//////////////////////////////////////*/

    /*//////////////////////////////////////*/
    printf("input for encrypt:\n");
    scanf ("%d",&m);
    encrypt("enc.txt", m);
    puts("------------------------------");
    /*//////////////////////////////////////*/


    /*//////////////////////////////////////*/
    printf("input for decrypt:\n");
    scanf ("%d",&m);
    decrypt ( "dec.txt", m);
    puts("------------------------------");
    /*//////////////////////////////////////*/

    /*//////////////////////////////////////*/
    third_largest();
    puts("------------------------------");
    /*//////////////////////////////////////*/

    return 0;
    /*END_OF_MAIN*/
```

```
}
double compute_serie( int m )
{
    int n;
    double sum;

    for(n=2;n<=m;++n)
        sum += pow(n,0.5)/(log(n)/log(2.0));

    return sum;
}
void draw_triangle(int size_triangle)
{
    int i,j,h;

    h = (size_triangle+1)/2;

    for(i=1;i<=h;++i) {
        for(j=1;j<=h-i;++j) {
            printf(" ");
        }
        for(j=1;j<=2*i-1;++j) {
            printf("*");
        }
    printf("\n");
    }
}
void draw_multi_triangle(int size_triangle,int count_triangle)
{
    int i;

    for(i=1;i<=count_triangle;i++)
        draw_triangle(size_triangle);

}
void draw_root(int size_root)
{
    int k,i,j;

    k=(size_root-1)/2;

    for(i=1;i<=size_root-k;i++){
        for(j=1;j<=k;j++)
            printf(" ");
        printf("*\n");
    }
    for(i=1;i<=size_root;i++)
        printf("*");


}
void draw_tree( int size_triangle,int count_triangle,int size_root)
{
    draw_multi_triangle(size_triangle,count_triangle);
    draw_root(size_root);
    printf("\n");

}
void encrypt(const char filename[], int n)
{
    int num;
    char c1,c2,c3;

    FILE *inp;
    inp=fopen( filename,"r");

    fscanf(inp,"%c%c%c%d",&c1,&c2,&c3,&num);

    /*C1 icin buyuk kucuk harf kontrolu*/

    if(c1 >= MIN && c1 <= MAX){
        c1 = (int)c1 + n;
```

```
        if( (int)c1 > MAX )
            c1 = MIN - 1 + (int)c1 - MAX;
    }
    if(c1 >= MIN2 && c1 <= MAX2){
        c1 = (int)c1 + n;
        if( (int)c1 > MAX2 )
            c1 = MIN2 - 1 + (int)c1 - MAX2;
    }

    /*C2 icin buyuk kucuk harf kontrolu*/

    if(c2 >= MIN && c2 <= MAX){
        c2 = (int)c2 + n;
        if( (int)c2 > MAX )
            c2 = MIN - 1 + (int)c2 - MAX;
    }
    if(c2 >= MIN2 && c2 <= MAX2){
        c2 = (int)c2 + n;
        if( (int)c2 > MAX2 )
            c2 = MIN2 - 1 + (int)c2 - MAX2;
    }

    /*C3 icin buyuk kucuk harf kontrolu*/
    if(c3 >= MIN && c3 <= MAX){
        c3 = (int)c3 + n;
        if( (int)c3 > MAX )
            c3 = MIN - 1 + (int)c3 - MAX;
    }

    if(c3 >= MIN2 && c3 <= MAX2){
        c3 = (int)c3 + n;
        if( (int)c3 > MAX2 )
            c3 = MIN2 - 1 + (int)c3 - MAX2;
    }




    printf("%c%c%c%d\n",c1,c2,c3,num);


}
void decrypt(const char filename[], int n)
{
    int num;
    char c1,c2,c3;

    FILE *inp;
    inp=fopen( filename,"r");

    fscanf(inp,"%c%c%c%d",&c1,&c2,&c3,&num);

    /*C1 icin buyuk kucuk harf kontrolu*/
    if(c1 >= MIN && c1 <= MAX) {
        c1 = (int)c1 - n ;
        if( (int)c1 < MIN )
            c1 = MAX + 1 - (MIN - (int)c1 );
    }
    if(c1 >= MIN2 && c1 <= MAX2){
        c1 = (int)c1 - n ;
        if( (int)c1 < MIN2 )
            c1 = MAX2 + 1 - (MIN2 - (int)c1 );
    }

    /*C2 icin buyuk kucuk harf kontrolu*/
    if(c2 >= MIN && c2 <= MAX) {
        c2 = (int)c2 - n ;
        if( (int)c2 < MIN )
            c2 = MAX + 1 - (MIN - (int)c2 );
    }
    if(c2 >= MIN2 && c2 <= MAX2){
        c2 = (int)c2 - n ;
```

```
        if( (int)c2 < MIN2 )
            c2 = MAX2 + 1 - (MIN2 - (int)c2 );
    }

    /*C3 icin buyuk kucuk harf kontrolu*/
    if(c3 >= MIN && c3 <= MAX) {
        c3 = (int)c3 - n ;
        if( (int)c3 < MIN )
            c3 = MAX + 1 - (MIN - (int)c3 );
    }
    if(c3 >= MIN2 && c3 <= MAX2){
        c3 = (int)c3 - n ;
        if( (int)c3 < MIN2 )
            c3 = MAX2 + 1 - (MIN2 - (int)c3 );
    }

    printf("%c%c%c%d\n",c1,c2,c3,num);


}
void third_largest()
{
    char c1,c2,c3,temp,skip,ch;

    printf("Enter a characters:\n");
    scanf(" %c%c%c%c%c%c",&c1,&skip,&c2,&skip,&c3,&skip);

    if(c1 > c2){
        temp = c1;
        c1 = c2;
        c2 = temp;
    }
    if(c2 > c3){
        temp = c2;
        c2 = c3;
        c3 = temp;
    }
    if(c1 > c2){
        temp = c1;
        c1 = c2;
        c2 = temp;
    }

    scanf("%c",&ch);

    while(ch != ' '){

        if(c3 < ch){
            c1 = c2;
            c2 = c3;
            c3 = ch;
        }
        else if(c2 < ch){
            c1 = c2;
            c2 = ch;
        }
        else if(c1 < ch)
            c1 = ch;

        scanf("%c",&ch);
    }

    printf("Third largest:%c\n",c1);

}
/*############################################################################*/
/*              End of HW04_121044074_Onur_Sezer                              */
/*############################################################################*/
```