```c
/*######################################################################*/
/*  HW08_HASAN_MEN_131044009_part1.c                                    */
/*  HAZIRLAYAN  : HASAN MEN - 131044009                                 */
/*  TARIH       : 20.4.15                                               */
/*                                                                      */
/*  TANIM       : Dosyadan okunan tablonun icinde icerik toplami        */
/*  maximum olan dikdortgeni origin ve rastgele noktalara gore          */
/*  bulan program parcacigi                                             */
/*                                                                      */
/*######################################################################*/
#include<stdio.h>

#define COL_COUNT 8
#define ROW_CAP 10  /*Maximum y ekseni sayisi*/


typedef struct
{
    int x;
    int y;
}Point_t;

/*Dikdortgen icerigi struct olacak*/
typedef struct
{
    Point_t left_up;
    Point_t right_down;
    double sum;
}Rectangle_t;

/*X ve ye noktasi alip bunlari bir kose noktasi haline getirir.*/
Point_t construct_point(int x,int y);

/*Dikdortgenin solust ve sagalt koselerini alip */
/*dikdortgeni temsil eden yapiyi return eder*/
Rectangle_t construct_rectangle(Point_t left_up,Point_t right_down);

/*Maximum icerikli dikdortgenin koselerini ekrana basar */
void print_rectangle(const Rectangle_t *rectangle);

/*Dosyadan dikdortgenin icerigini okuruz*/
void getArray(FILE* inFile, double table[][COL_COUNT], int* nRow);

/*Belirli koseler arasindaki elemanlar toplamini bulur */
void getSum(double table[][COL_COUNT],Rectangle_t *rectangle);

/*Baslangic noktasi verilen dikdortgen icin maximum toplami bulur*/
Rectangle_t maxSumConstPoint(   double table[][COL_COUNT],
                                int nRow, Point_t left_up);

/*Tum koseleri deneyerek maximum toplami bulur */
/*degerleri dikdortgen turunden return eder*/
Rectangle_t maxSumRec(double table[][COL_COUNT], int nRow);

int main(){
    double table[ROW_CAP][COL_COUNT];   /*Degerlerin okunacagi dizi*/
    FILE* inFile;
    int nRow;   /*Satir sayisi - yekseni sayisi*/

    Rectangle_t rectangle;  /*Return edilen dikdortgenler icin temp*/
    Point_t start=construct_point(0,0); /*maxSumConstpoint icin start noktasi*/

    inFile=fopen("Table1.txt","r"); /*input dosyamiz*/

    /*Dosyadan degerler okundu...*/
    getArray(inFile, table, &nRow);

    /*Origine gore degerler bulundur ve return edilip ekrana basildis*/

    rectangle = maxSumConstPoint(table, nRow,start);
    print_rectangle(&rectangle);

    /*Tum olasiliklara bakilarak en buyuk toplam bulundu*/
    /*ve degerler ekrana basildi...*/
    rectangle = maxSumRec(table,nRow);
    print_rectangle(&rectangle);

    /*input dosyasi kapandi ve ana fonksiyon bitti*/
    fclose(inFile);
    return 0;
}

/* Dosyadaki degerlerin 2D arraya doldurulmasi*/
void getArray(FILE* inFile, double table[][COL_COUNT], int* nRow)
{
    int row=0;
    int col;
    int status=EOF+1;/*Different from EOF*/

/*Dongude 1satir fazla okuyacak onu dikkate alarak degerler return edilmeli*/
    while(status!=EOF){
        for(col=0; col<COL_COUNT; col++)
           status=fscanf(inFile, "%lf", &table[row][col]);
        ++row;
    }
    *nRow=row-1;/*Fazla satir silindi*/

}

/*Belirli koseler arasindaki elemanlar toplamini bulur */
void getSum(double table[][COL_COUNT],Rectangle_t *rectangle)
{

    int row, col;
    (*rectangle).sum=0;
    for(row=(*rectangle).left_up.y; row<=(*rectangle).right_down.y; ++row)
        for(col=(*rectangle).left_up.x; col<=(*rectangle).right_down.x; ++col)
                ((*rectangle).sum)+=table[row][col];
}

/*Baslangic noktasi verilen dikdortgen icin maximum toplami bulur*/
Rectangle_t maxSumConstPoint(   double table[][COL_COUNT],
                                int nRow, Point_t left_up)
{
    int temp=0;
    Rectangle_t mSCP;   /* dikdortgen bilgileri icin yer*/

    Point_t rDown=construct_point(left_up.x,left_up.y); /* alt kosemiz */
    Point_t find;   /* max deger bulununca stoklanacak yer */

    /* rectannlge baslangic icin dolduruldu */
    mSCP.sum=table[left_up.x][left_up.y];
    mSCP=construct_rectangle(rDown,rDown);

    /*alt kose degistirilerek tum olasiliklar denenir*/
    for(rDown.y=left_up.y; rDown.y<nRow; ++rDown.y){
        for(rDown.x=left_up.x; rDown.x<COL_COUNT; ++rDown.x)
            {
                /* alt kose belirlendi ve toplamlar bulundu */
                mSCP.right_down=construct_point(rDown.x,rDown.y);
                getSum(table,&mSCP);
                if(mSCP.sum>temp)
                {
                /* max toplam bulununca degerleri depolariz */
                    temp=mSCP.sum;
                    find=construct_point(rDown.x,rDown.y);
                }
            }
    }
    /* bulunan degerlerin rectangle ye yazilmasi ve return edilmesii */
    mSCP.sum=temp;
    mSCP.right_down=construct_point(find.x,find.y);

    return mSCP;
```

```c
}

Point_t construct_point(int x,int y)
{
    Point_t points;

    points.x=x;
    points.y=y;

    return points;
}

Rectangle_t construct_rectangle(Point_t left_up,Point_t right_down)
{
    Rectangle_t c_rec;

    c_rec.left_up=left_up;
    c_rec.right_down=right_down;
    return c_rec;
}
Rectangle_t maxSumRec(double table[][COL_COUNT], int nRow)
{
    Point_t start;  /* baslangic noktalari -  sol ust */
    Rectangle_t new_rec;    /* maxSumconst poinnten gelen rect icin yer*/
    Rectangle_t found;  /* maximum icerikli rectangle */
    double temp=0;
    int lUY, lUX;   /*coordinates of the left upper corner*/
    /*initialize the rectangular with the one including only origin point*/


    /*For all feasible starting points call maxSumConstPoint*/
    for(lUY=0; lUY<nRow; lUY++){
        for(lUX=0; lUX<COL_COUNT;lUX++)
        {
        /* baslangic noktalari belirlenir ve her biri icin rectangle bulunur*/
            start=construct_point(lUX,lUY);
            new_rec=maxSumConstPoint(table, nRow,start);

            if(new_rec.sum>temp)
            {
                /* maximimum rectangle bulununca founda yazilir*/
                temp=new_rec.sum;
                found.sum=temp;
                found = construct_rectangle(new_rec.left_up,new_rec.right_down);
            }
        }
    }
    return found;
}

void print_rectangle(const Rectangle_t *rectangle)
{
    printf("MaxSum Rectangular starting from (Y=%d,X=%d) is %.2f.\n",
                                            (*rectangle).left_up.y,
                                            (*rectangle).left_up.x,
                                            (*rectangle).sum);
    printf("Its right down coordinate (Y=%d,X=%d)\n",
                                (*rectangle).right_down.y,
                                (*rectangle).right_down.x);
    printf("#####################################################\n");


}

/* HW08_HASAN_MEN_131044009_part1.c  SONU*/
```