```
/*##########################################################################*/
/*HW08_part1.c                                                              */
/*_______________________________                                           */
/*Written by Mustafa Akilli on April 18, 2015                               */
/*                                                                          */
/*Description                                                               */
/*___________                                                               */
/*  Max Rectangular Sum Problem                                             */
/*Inputs:                                                                   */
/*  -Table.txt                                                              */
/*Outputs:                                                                  */
/*  -The Max Sum Rectangle starting from origin                             */
/*  -The Max Sum Rectangle                                                  */
/*##########################################################################*/
/*                                                                          */
/*--------------------------------------------------------------------------*/
/*                           Includes                                       */
#include <stdio.h>
/*--------------------------------------------------------------------------*/
/*                            Define                                        */
#define COL_COUNT 8
#define ROW_CAP 10
/*--------------------------------------------------------------------------*/
/*                            Structure                                     */
typedef struct
{
    int x;
    int y;
} Point_t;


typedef struct
{
    Point_t left_up;
    Point_t right_down;
    double sum;
}Rectangle_t;
/*--------------------------------------------------------------------------*/
/*                            Functions                                     */
Point_t construct_point(int x, int y);
Rectangle_t construct_rectangle(Point_t left_up, Point_t right_down);
void print_rectangle(const Rectangle_t *rectangle);
void getArray(FILE* inFile, double table[][COL_COUNT], int* nRow);
void getSum(double table[][COL_COUNT], Rectangle_t *rectangle);
Rectangle_t maxSumConstPoint(double table[][COL_COUNT], int nRow, Point_t
left_up);
Rectangle_t maxSumRec(double table[][COL_COUNT], int nRow);
/*--------------------------------------------------------------------------*/


int
main(void){

    double table[ROW_CAP][COL_COUNT];
    int nRow;
    Point_t left_up,right_down;
    Rectangle_t rectangle;

    FILE* inFile;

    inFile=fopen("Table1.txt", "r");

    left_up=construct_point(0,0);

    rectangle=construct_rectangle(left_up,right_down);


    getArray(inFile, table, &nRow);

    rectangle=maxSumConstPoint(table,nRow,rectangle.left_up);

    printf("\nMaxSum Rectangular starting from origin is %.2f.\n",rectangle.sum);
```

```c
    printf("Its right down coordinate (y,x) is ");
    printf("%d, %d\n",rectangle.right_down.y, rectangle.right_down.x);

    print_rectangle(&rectangle);

    rectangle=maxSumRec(table,nRow);

    printf("MaxSum Rectangular is %.2f.\n",rectangle.sum);
    printf("Its left uppercoordinate (y,x) is %d",rectangle.left_up.y);
    printf(",%d,\nright down coordinate is ",rectangle.left_up.x);
    printf("%d, %d\n",rectangle.right_down.y,rectangle.right_down.x);

    print_rectangle(&rectangle);

    fclose(inFile);

    return 0;
}

/* Takes 2 integers, returns a Point_t representing these integers.             */
Point_t construct_point(int x, int y)
{
    Point_t representing_integers;

    representing_integers.x=x;
    representing_integers.y=y;

    return representing_integers;
}

/* Takes 2 points, returns a Rectangle_t representing these points.             */
Rectangle_t construct_rectangle(Point_t left_up, Point_t right_down)
{
    Rectangle_t representing_points;

    representing_points.left_up=left_up;
    representing_points.right_down=right_down;

    return representing_points;
}

/* takes a rectangle pointer and prints all information about it in a reasonable
    format.                                                                     */
void print_rectangle(const Rectangle_t *rectangle)
{
    printf("\nrectangle.left_up.y:%d\n",rectangle->left_up.y);
    printf("rectangle.left_up.x:%d\n",rectangle->left_up.x);
    printf("rectangle.right_down.y:%d\n",rectangle->right_down.y);
    printf("rectangle.right_down.x:%d\n",rectangle->right_down.x);
    printf("rectangle.sum:%.2f\n\n",rectangle->sum);
}

/* Reads the table from a file into a 2D array                                  */
void getArray(FILE* inFile, double table[][COL_COUNT], int* nRow)
{
    int row=0;
    int col;
    int status=EOF+1;/*Different from EOF*/

    /*one more row will be read but the values will not be recorded into the
    table therefore, it is safe to use a table having just enough capasity
    to hold the data*/
    while(status!=EOF){
        for(col=0; col<COL_COUNT; col++)
            status=fscanf(inFile, "%lf", &table[row][col]);
        ++row;
    }

    *nRow=row-1;/*one more row read*/
}

/* Returns the sum inside a given rectangular                                   */
```

```
void getSum(double table[][COL_COUNT], Rectangle_t *rectangle)
{
    int row, col;
    double 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)
        {
            sum+=table[row][col];
        }

    rectangle->sum=sum;
}

/*Finds the rectangular left uppper point of which is specified having the
max sum inside                                                              */
Rectangle_t maxSumConstPoint(double table[][COL_COUNT], int nRow, Point_t
left_up)
{
    int rDX;    /*x coordinate of the right down corner of the rec*/
    int rDY;    /*y coordinate of the right down corner of the rec*/
    double temp;
    /*initialize the rectangular with the one including only one point*/
    double sum=table[left_up.x][left_up.y];
    Point_t right_down;
    Rectangle_t rectangle;
    Rectangle_t temp_rectangle;

    rectangle.left_up=left_up;

    rectangle.right_down.y=rectangle.left_up.y;
    rectangle.right_down.x=rectangle.left_up.x;

    temp_rectangle.left_up.y=rectangle.left_up.y;
    temp_rectangle.left_up.x=rectangle.left_up.x;

    /*Try all feasible rectangulars by changing the right down corner*/
    for(rDY=rectangle.left_up.y; rDY<nRow; ++rDY){
        for(rDX=rectangle.left_up.x; rDX<COL_COUNT; ++rDX){

            temp_rectangle.right_down.y=rDY;
            temp_rectangle.right_down.x=rDX;



            getSum(table,&temp_rectangle);
            temp=temp_rectangle.sum;
            if(temp>sum){
                /*a better rectangular is found, perform an update */
                sum=temp;
                rectangle.right_down.y=rDY;
                rectangle.right_down.x=rDX;
            }
        }
    }

    rectangle.sum=sum;


    return (rectangle);
}

Rectangle_t maxSumRec(double table[][COL_COUNT], int nRow)
{

    double temp;
    int lUY, lUX;   /*coordinates of the left upper corner*/
    int rDY, rDX;   /*coordinates of the right down corner*/
    /*initialize the rectangular with the one including only origin point*/
    double maxSum=table[0][0];
    Rectangle_t rectangle,temp_rectangle;
    rectangle.left_up.y=0;
    rectangle.left_up.x=0;
    rectangle.right_down.y=0;
    rectangle.right_down.x=0;

    /*For all feasible starting points call maxSumConstPoint*/
    for(lUY=0; lUY<nRow; ++lUY){
        for(lUX=0; lUX<COL_COUNT; ++lUX){

            temp_rectangle.left_up.y=lUY;
            temp_rectangle.left_up.x=lUX;

            temp_rectangle=maxSumConstPoint(table,nRow,temp_rectangle.left_up);
            temp=temp_rectangle.sum;
            if(temp>maxSum){
                /*a better rectangular found, perform an update*/
                maxSum=temp;
                rectangle.left_up.y=lUY;
                rectangle.left_up.x=lUX;
                rectangle.right_down.y=temp_rectangle.right_down.y;
                rectangle.right_down.x=temp_rectangle.right_down.x;
            }
        }
    }

    rectangle.sum=maxSum;

    return rectangle;
}
/*###############################################################################*/
/*                            End of HW08_part1.c                                */
/*###############################################################################*/
```