

```

#include<stdio.h>

#define COL_COUNT 8
#define ROW_CAP 10

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

typedef struct { Point_t left_up;
                Point_t right_down;
                double sum;
            } Rectangle_t;

Point_t constrcut_point(int x , int y);
Rectangle_t constrcut_rectangle(Point_t left_up, Point_t right_down);
void print_rectangle(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, const Point_t left_up);
Rectangle_t maxSumRec(double table[][COL_COUNT], int nRow);

int main(){
    double table[ROW_CAP][COL_COUNT];
    FILE* inFile;
    int nRow;
    Rectangle_t rectangle;
    rectangle.left_up.x = 0;    rectangle.left_up.y = 0;
    rectangle.right_down.x = 0; rectangle.right_down.y = 0;
    rectangle.sum = 0;

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

    getArray(inFile, table, &nRow);

    rectangle=maxSumConstPoint(table, nRow, rectangle.left_up);
    printf("MaxSum Rectangular starting from origin is %.2lf. Its right down coordinate (y,x) is %d, %d\n", rectangle.sum, rectangle.right_down.y, rectangle.right_down.x);

    rectangle=maxSumRec(table, nRow);
    printf("MaxSum Rectangular is %.2lf. Its left upper coordinate (y,x) is %d, %d, right down coordinate is %d, %d\n", rectangle.sum, rectangle.left_up.y, rectangle.left_up.x, rectangle.right_down.y, rectangle.right_down.x);

    fclose(inFile);
    return 0;
}

/*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 capacity 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;
        (*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];
    }

/*Finds the rectangular left upper point of which is specified having the max sum inside*/
Rectangle_t maxSumConstPoint(double table[][COL_COUNT], int nRow, const Point_t left_up)
{
    Rectangle_t temp;
    Rectangle_t rectangle;
    /*initialize the rectangular with the one including only one point*/
    temp.left_up=left_up;

    rectangle.sum=table[left_up.x][left_up.y];

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

    /*Try all feasible rectangulars by changing the right down corner*/
    for(temp.right_down.y=left_up.y; temp.right_down.y<nRow; ++temp.right_down.y){
        for(temp.right_down.x=left_up.x; temp.right_down.x<COL_COUNT; ++temp.right_down.x){
            getSum(table, &temp);
            if(temp.sum>rectangle.sum){
                /*a better rectangular is found, perform an update */
                rectangle=temp;
            }
        }
    }

    return rectangle;
}

```

```

Rectangle_t maxSumRec(double table[][COL_COUNT], int nRow)
{
    Rectangle_t temp;
    Point_t leftup;
    /*initialize the rectangular with the one including only origin point*/
    Rectangle_t max;
    max.sum=table[0][0];
    max=constrcut_rectangle(constrcut_point(0,0),constrcut_point(0,0));

    /*For all feasible starting points call maxSumConstPoint*/
    for(leftup.y=0; leftup.y<nRow; leftup.y+=1){
        for(leftup.x=0; leftup.x<COL_COUNT; leftup.x+=1){
            temp=maxSumConstPoint(table, nRow, leftup);
            if(temp.sum>max.sum){
                /*a better rectangular found, perform an update*/
                max=temp;
            }
        }
    }

    return max;
}

```

```

Point_t constrcut_point(int x , int y)
{
    Point_t point;

    point.x = x;
    point.y = y;

    return point;
}

```

```

Rectangle_t constrcut_rectangle(Point_t left_up, Point_t right_down)
{
    Rectangle_t rect;

    rect.left_up = left_up;
    rect.right_down = right_down;
    rect.sum=0;

    return rect;
}

```

```
}  
void print_rectangle(Rectangle_t *rectangle)  
{  
    printf("left_up { x = %d, y = %d}\n right_down { x = %d, y = %d}\n sum=%lf\n",  
          (*rectangle).left_up.x, (*rectangle).left_up.y,  
          (*rectangle).right_down.x, (*rectangle).right_down.y, (*rectangle).sum);  
}
```