

Mentor, A Siemens

D2S calibre internship qualifications
C++ MINI PROJECT
REPORT



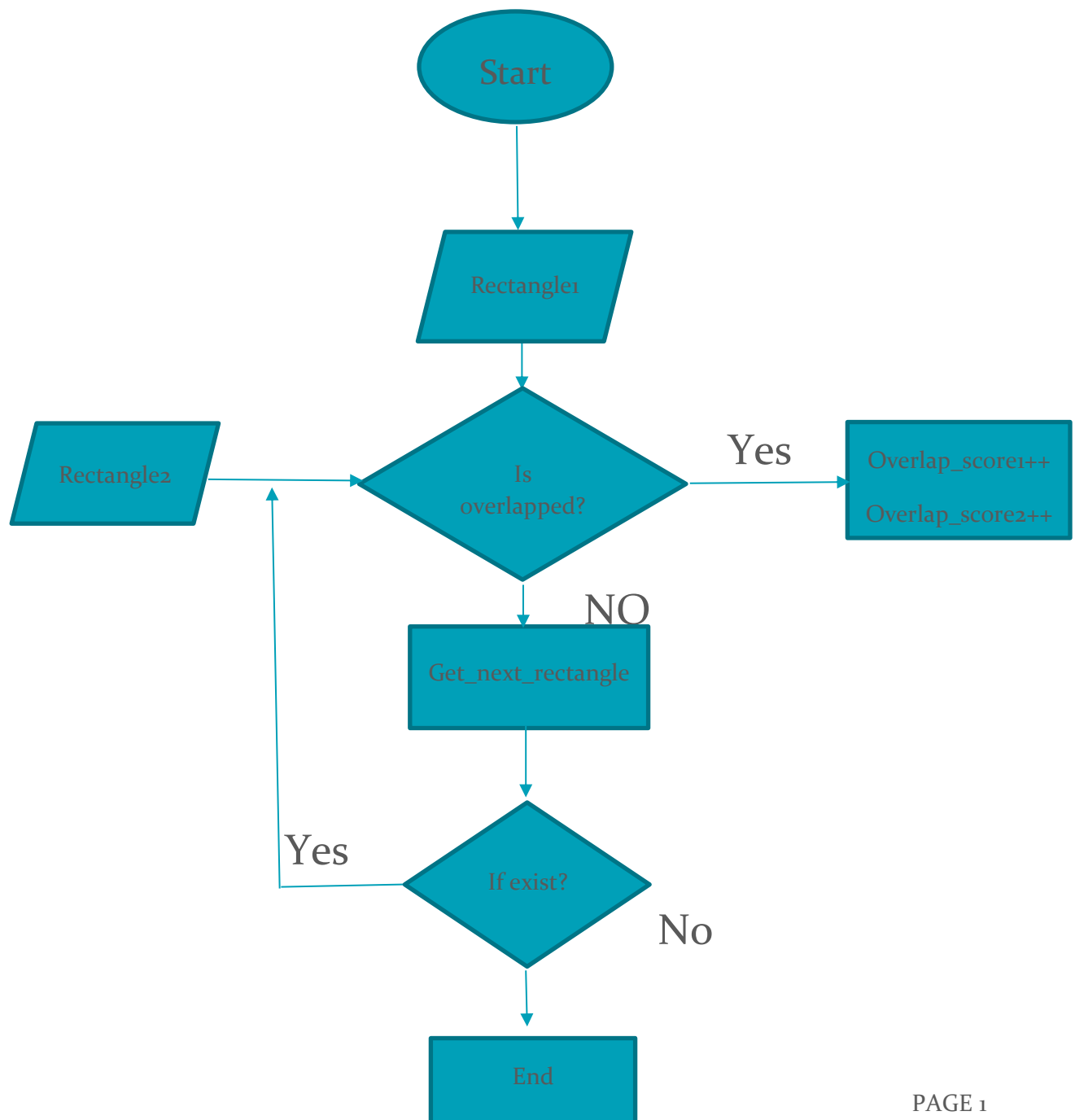
NAME: ESLAM TAREK FAWZY

Email: eslamsolari1996@gmail.com

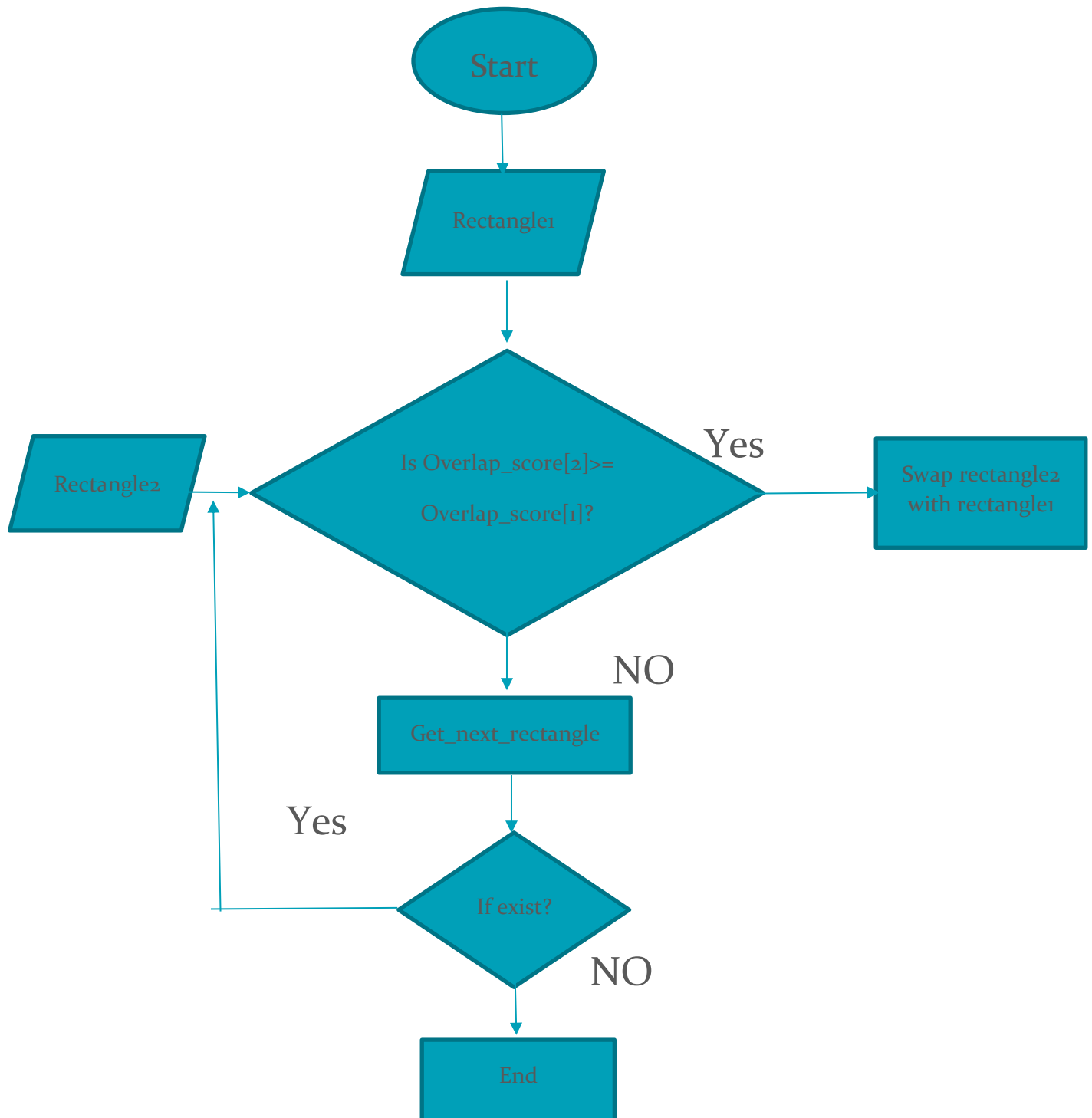
-**Platform:** Linux (Ubuntu 16), Clion IDE, also I used visual studio209 tool to get memory usage using performance profiler

-**Flow charts:**

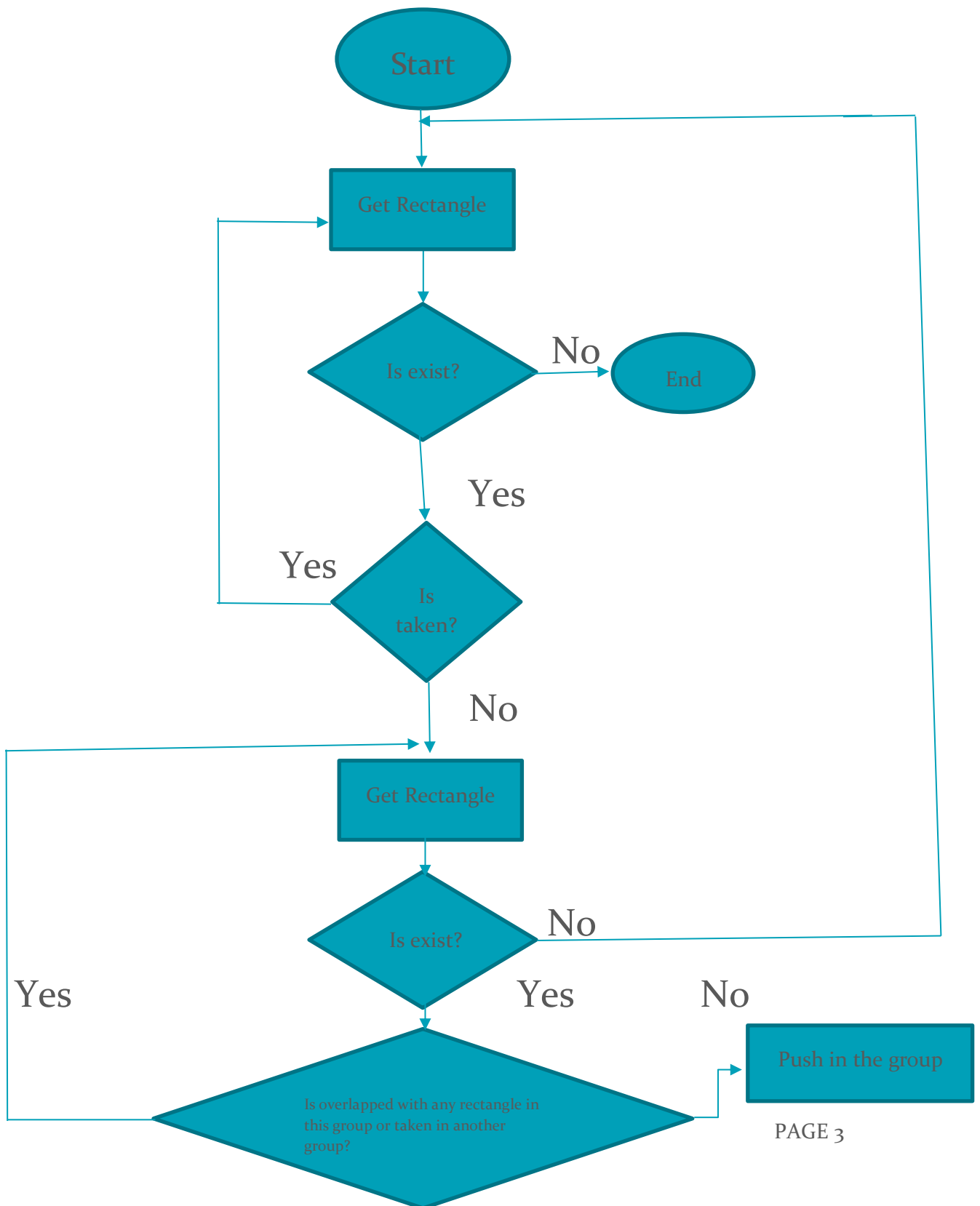
Calculating overlapping score of each rectangle



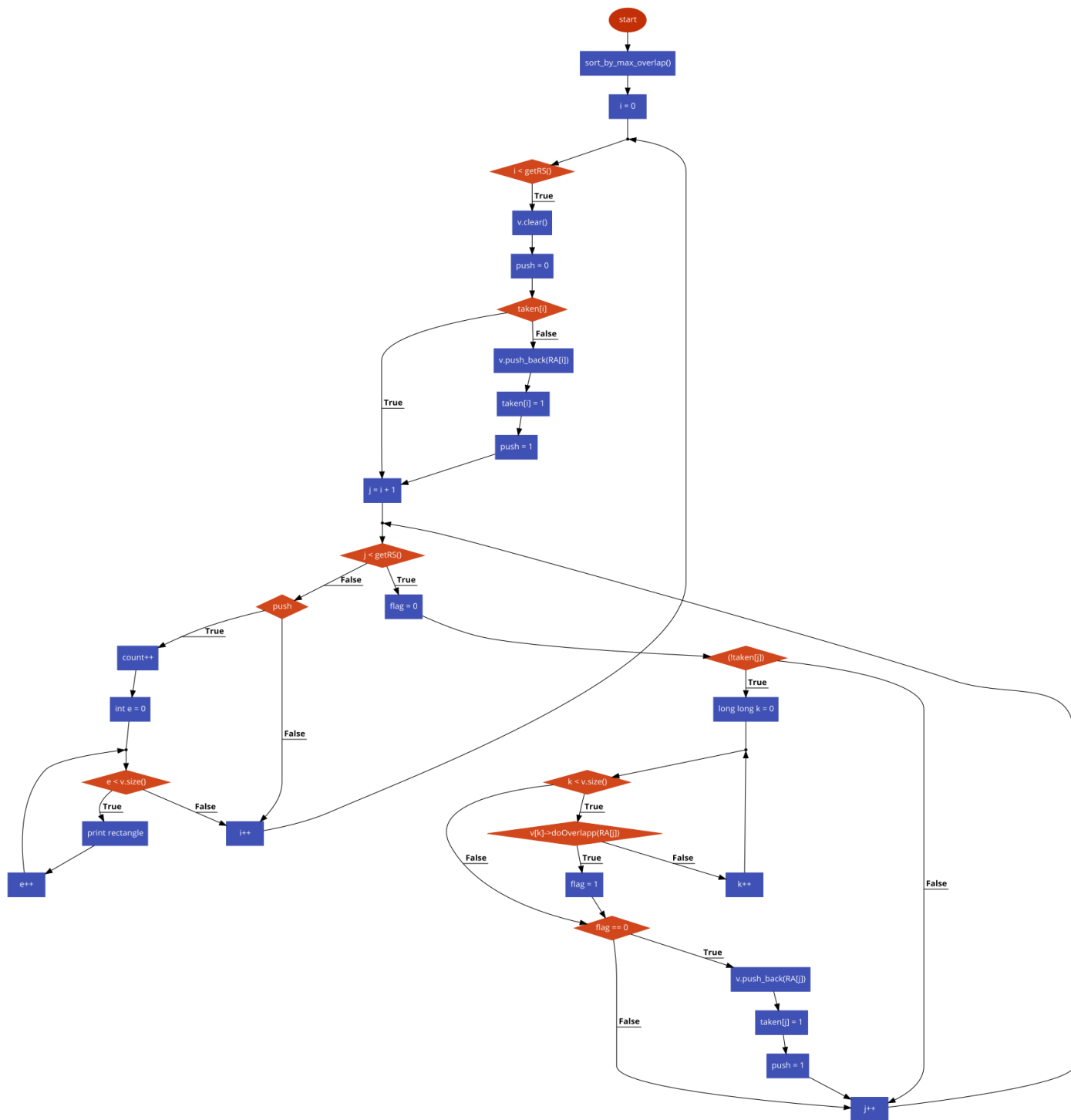
Sorting rectangles according to overlapping score



After sorting rectangles according to scores get
minimum number of groups



Flow chart using Code2flow:



-Pseudo Code

```
int overlap[number_rectangles] //initially equal zero

For i=0 ; i<number_rectangles; i++ :

    For j=0 ; j<number_rectangles; j++ :

        If rectangle[i] is_overlapped(rectangle[j]):

            overlap[i]++

            overlap[j]++

for i = 0; i < number_rectangles ;i++:
    for long long j = i + 1; j < getRS(); j++:
        if (overlap[j] >= overlap_score[i])
            swap(rectangle[i],rectangle[j])

rectangle v[]; // indicate current group rectangles
bool taken[number_rectangles] //initially zero indicate if rectangle is in group or not
flag=0 // to break and get next_rectangle if found that current rectangle is taken or overlapping
with another rectangle in current group
push=0 // indicate if a new group is created
count=0; // minimum number of groups
    for i = 0; i < number_rectangles; i++
        v.clear();//remove all elements in v at start of new group
        push = 0
        if (!taken[i]):
            v.push(rectangle[i])
            taken[i] = 1
            push = 1
        for j = i + 1; j < getRS(); j++:
            flag = 0
            if (!taken[j]): //if rectangle is not group
                for k = 0; k < v.size(); k++:
```

```
// and not overlapping with another rectangle in this group
```

```
        if v[k] is_overlapped(rectangle[j]):  
            flag = 1  
            break  
    }  
    if (flag == 0) { // indicate that this rectangle is not in another  
group and not overlapping with any rectangle in current group  
        v.push (rectangle[j])  
        taken[j] = 1  
        push = 1  
    }  
    if (push) :  
        count++;
```

Note: I tried to sort the rectangle at first by their x,y values if it would give better solution and minimum number of groups.

The number of output groups, run time elapsed to finish and the memory usage

Data Set	Number of Inputs	Number of output groups	Runtime (sec)	Memory usage (MB)
data_set_1.txt	5	2	0.00218	0.5
data_set_2.txt	7	2	0.001162	0.5
data_set_3.txt	20	5	0.002686	0.65
data_set_4.txt	39	3	0.002153	0.657
data_set_5.txt	77	3	0.00367	0.71
data_set_6.txt	136	5	0.005675	0.768
data_set_7.txt	216	6	0.007551	0.82
data_set_8.txt	460	6	0.015804	0.852
data_set_9.txt	741	12	0.024395	0.912
data_set_10.txt	981	7	0.053718	1
data_set_11.txt	5793	8	1.126253	2.5
data_set_12.txt	6775	7	1.495119	2.8
data_set_13.txt	7538	7	1.829349	3.1
data_set_14.txt	8774	7	2.447906	3.4
data_set_15.txt	9188	7	2.954575	3.5
data_set_16.txt	25263575			

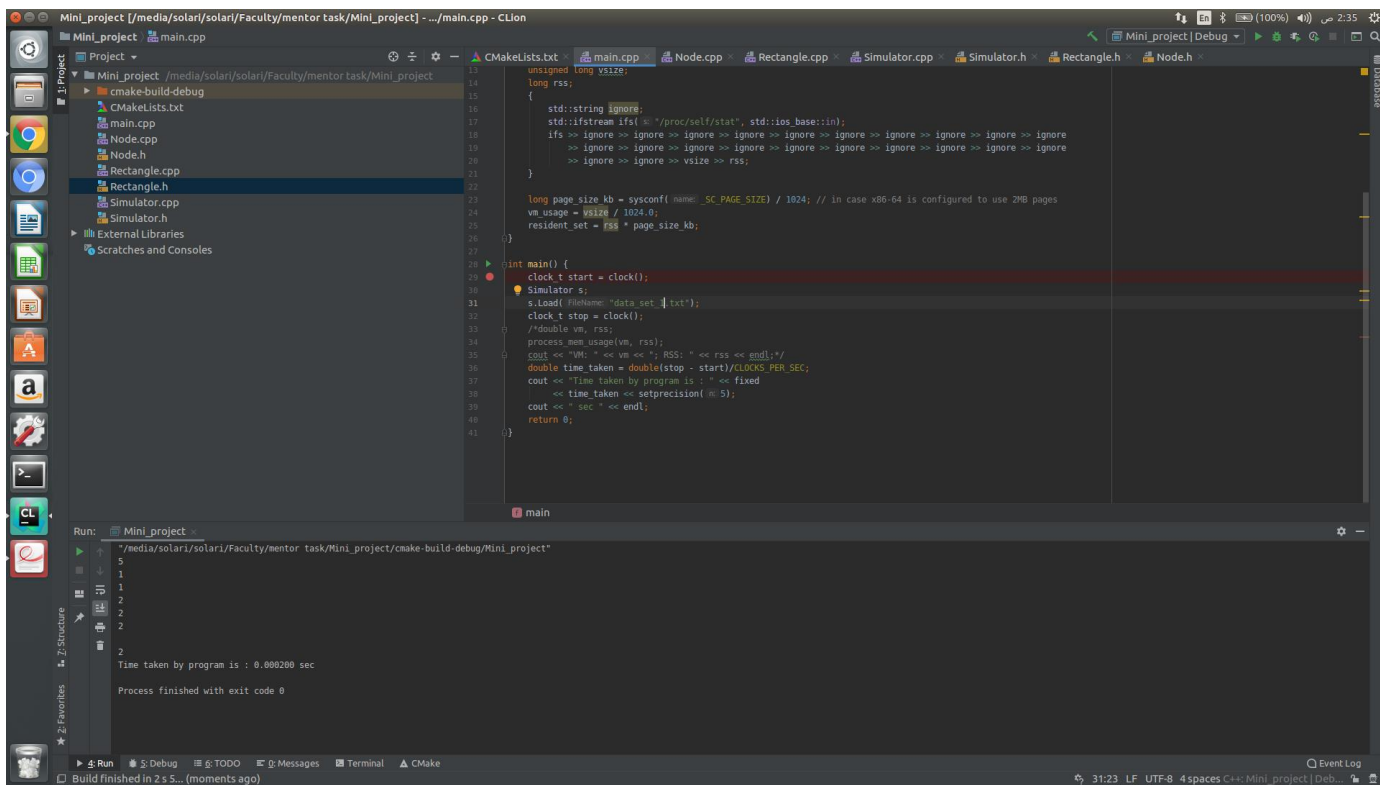
My algorithm was $O(N^2)$ and data_set_16 has about 25 million rectangles so it may take weeks to get the output and need higher HW.

Testing Methodology

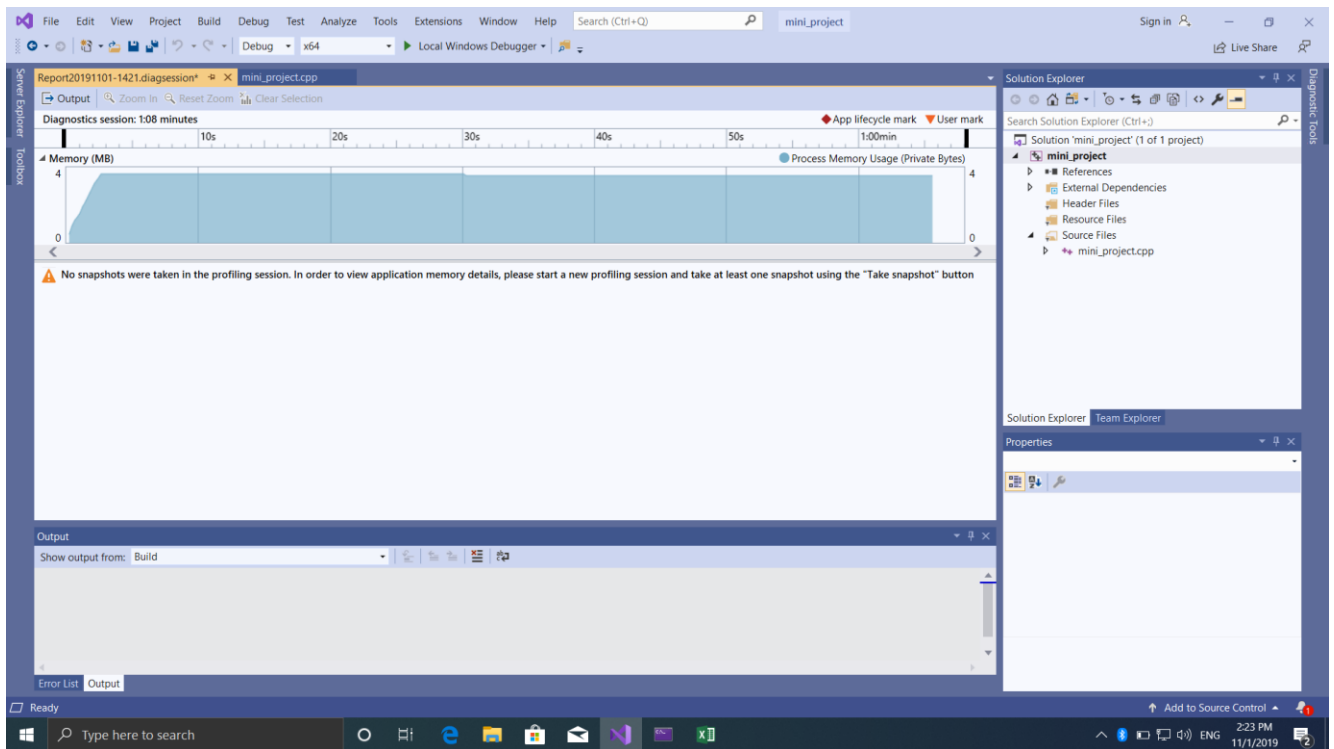
I have made testcases with myself and drew it using online drawing tools and get minimum number of groups and compare it with output of my program and tried many tricks.

Screen shots for the output of your program

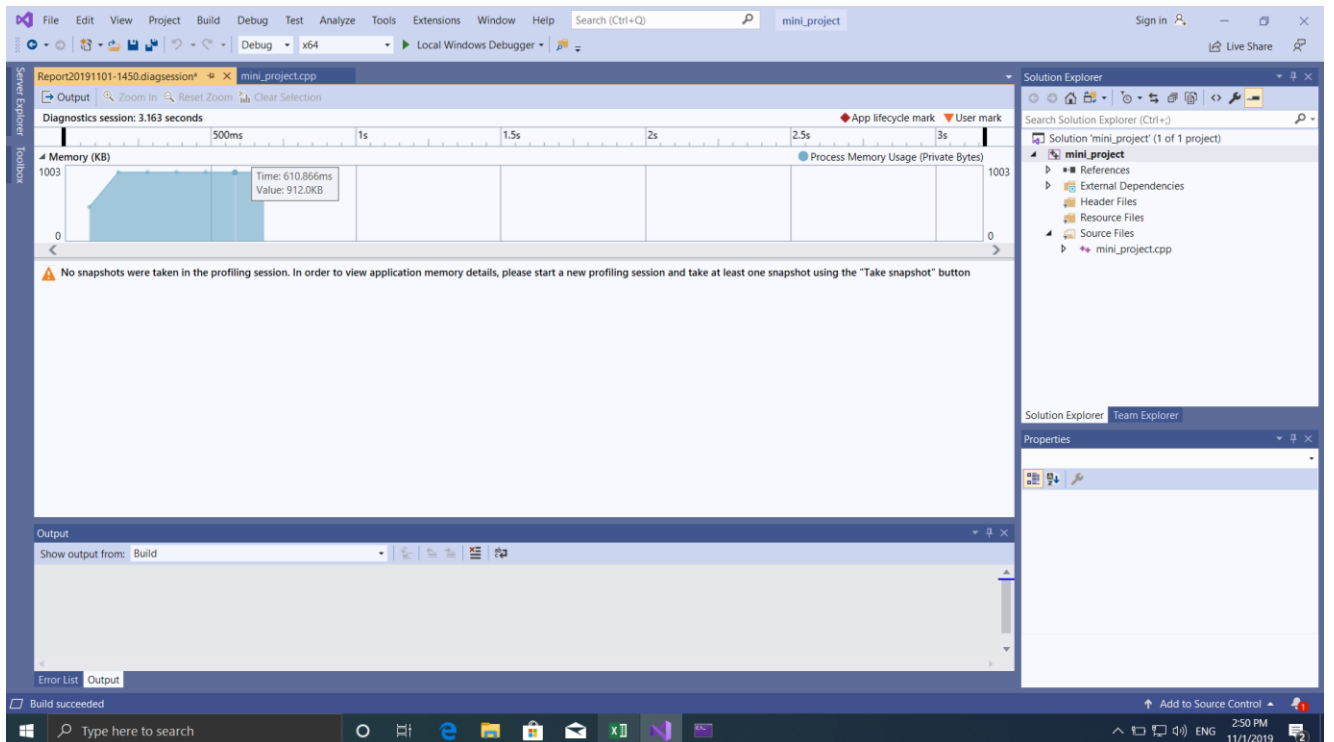
Printing overlap score of each rectangle in data_set_1



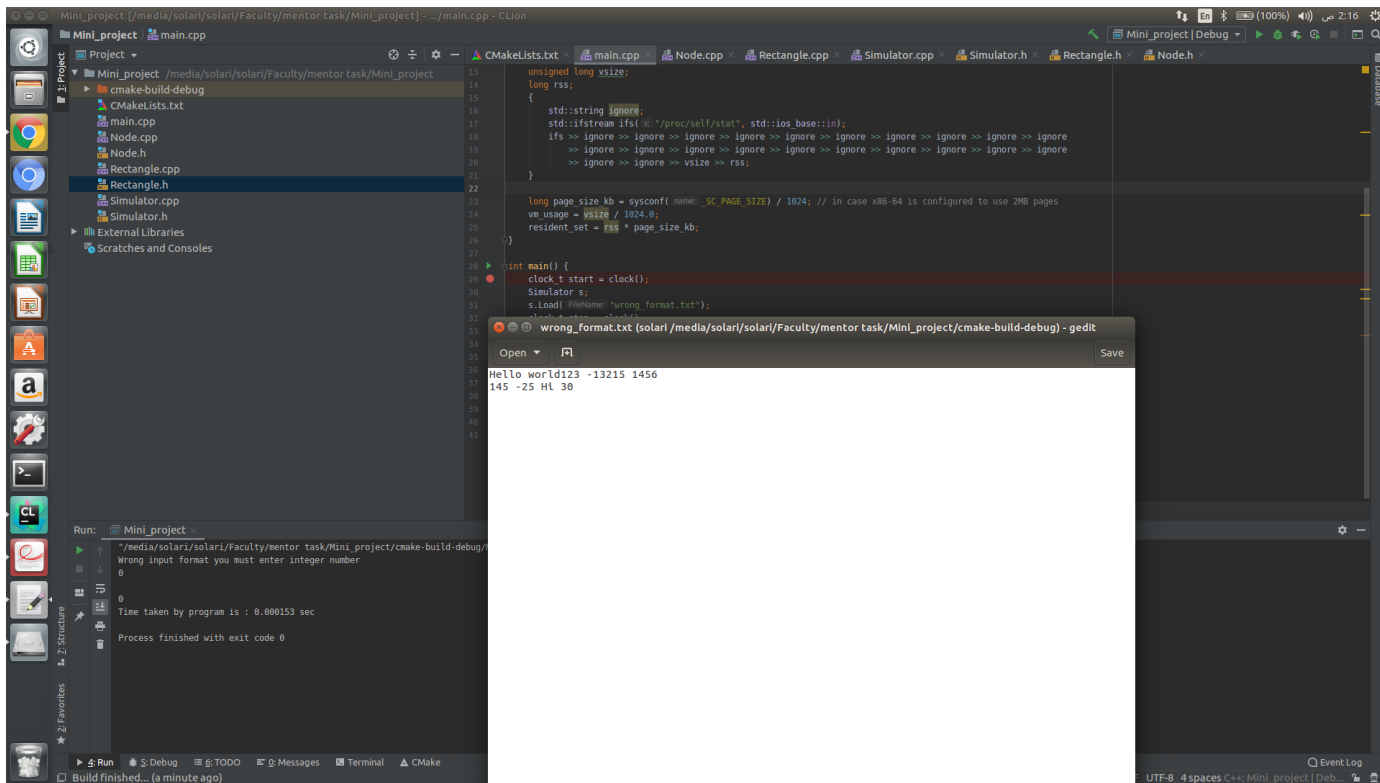
Memory usage using visual studio for data_set_15:



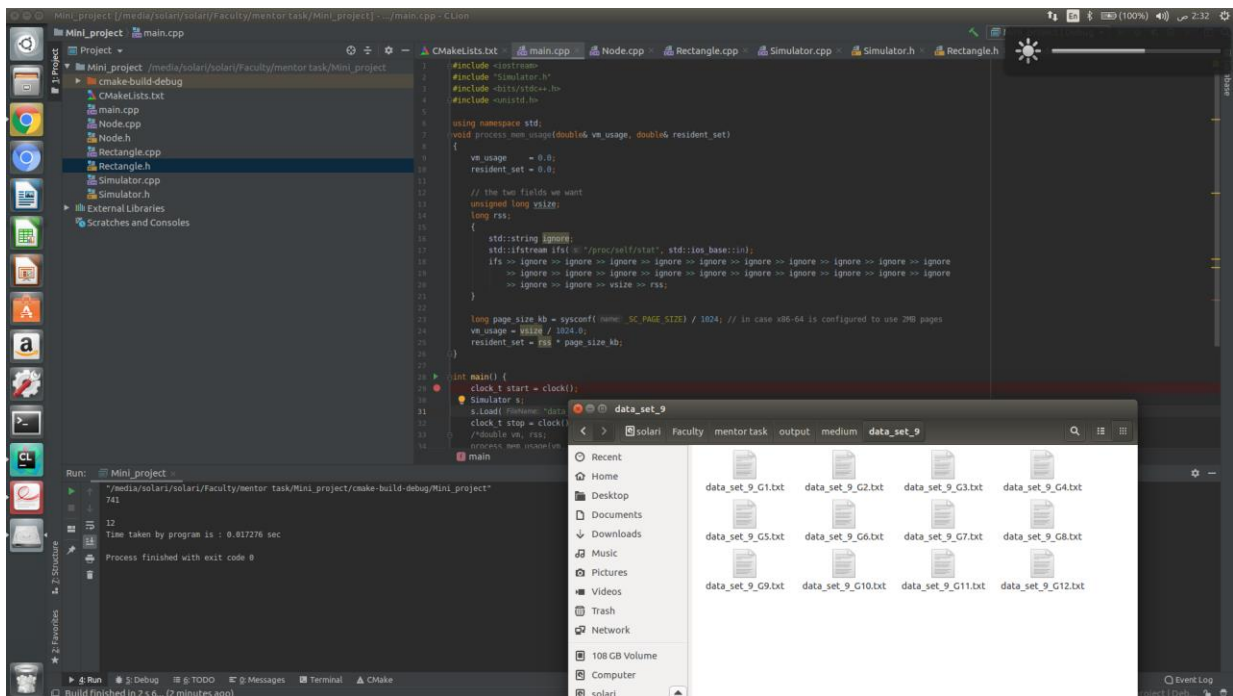
For data_set_9:



Handling wrong input format using exceptions:



Showing output files for data_set_9:



Showing content of output files for data_set_1:

