COP 3502C Spring 2020
Final term Assignment 1
Total points: 13

Introduction: For this assignment you have to write a c program that will utilize the merge sort and binary search algorithm.

What should you submit?
Write all the code in a single .c file and upload the .c file.

Please include the following lines in the beginning of your code to declare that you are the author of the code:

/* COP 3502C Final term Assignment 1
This program is written by: Your Full Name */

Compliance with Rules: UCF Golden rules apply towards this assignment and submission. Assignment rules mentioned in syllabus, are also applied in this submission.

Do not post or share part or full solution of your assignment anywhere!

Additional notes: The TAs and course instructor can call any students to explain their code for grading.

Problem Scenario

Jeff planned to travel some places during his spring break. However, due to the recent outbreak of the novel Coronavirus, he has canceled his travel plan and decided to stay home. However, as he enjoys coding, he wants spend some part his break on coding. He has received a file with several points (x,y) where the virus cases are found. He wants to write a code that can filter out the points closer to him and then he wants to make an efficient search system that can search a point (x,y) from the filtered list. He knows that binary search runs in O($\log_2$ n) which is much faster than a linear search. However, in-order to apply binary search, he needs to sort the points first. He learned about Merge Sort Algorithm recently and decided to apply Merge sort algorithm to sort the points of interest.

Problem formulation:

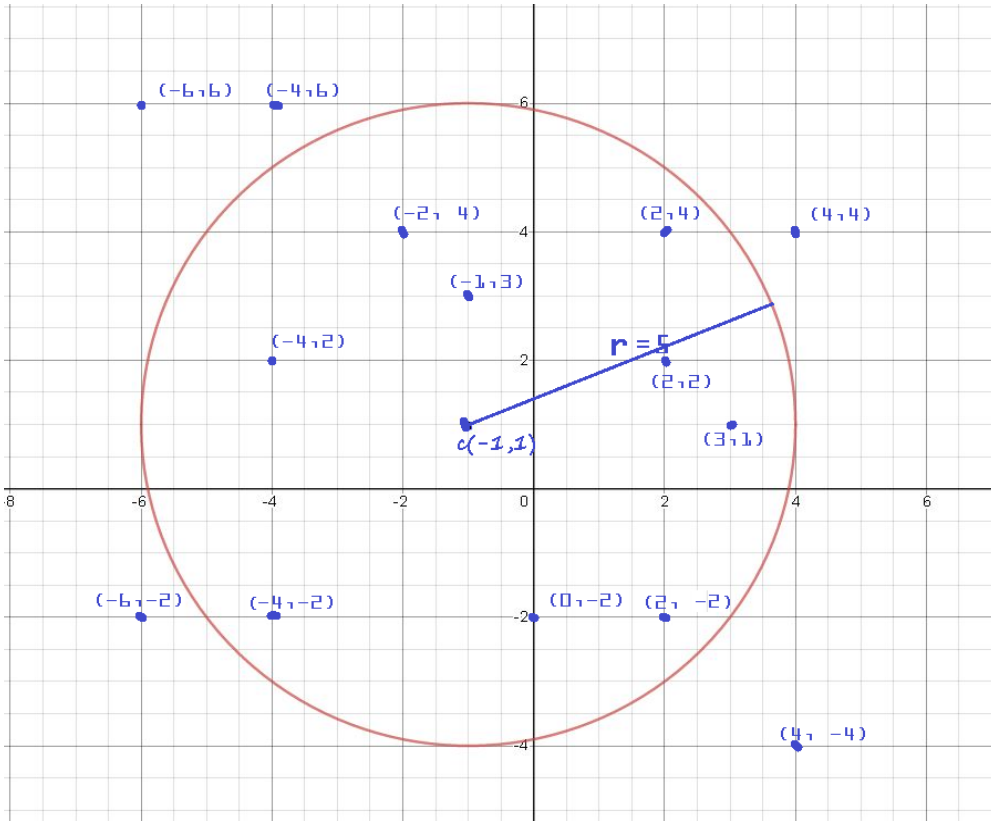
Given the center coordinate point (x,y) and the radius of a circle. Then there are N numbers of coordinate points also provided as part of the data. You have to filter out all the points those are inside or on the line of the circle. Next, you have to apply merge sort to sort those filtered points in x-axis major order. If two points have same x, then you should sort them based on y. After sorting, your program should continuously prompt the user for an input coordinate point x and y values and your program should perform binary search to determine whether the point exists in the given list of points or not. If the point is found, the program should show the position of the number in the filtered sorted list. The program stops taking further search key input if the provided value of x is -999 or y is -999. Keep reading the next paragraphs for better understanding.

Required Input/Output format:

The inputs related to the circle and points are taken from a text file called in.txt (create your own example in.txt file or copy them from the example bellow). You have to read the text file (in.txt) for reading the points. The first line of the file contains 4 numbers separated by space. The numbers are x coordinate of the center of the circle, y coordinate of the center of the circle, r the radius of the circle, and N that says how many points are there in the file. The next N lines of the file contain x and y coordinate values. Note that all the numbers in this file are integer. They can be both negative and positive numbers (except the radius). After reading the data, you have to filter all the points inside or on the line of the circle. Let us call it filtered list. Next, you have to sort those filtered points in x-axis major order *(it means that the data will be sorted based on x first, if multiple point has same x, then it will sort based on y)* and then write the output into another file (out.txt) in the same format, i.e., each line contains one coordinate point x and y and they are space delimited. After writing the sorted points in the file, your program should display the message “filtered and sorted data written to out.txt”. After that your program should continuously prompt the user for an input coordinate point x and y values and your program should perform binary search to determine whether the point exists in the given list of points. If it is found, it should show “Found” and record number within the filtered list. The program will stop prompting for x and y, if the search input x=-999 or y=-999

Example

Consider the circle and the points in the figure. The center of the circle is (-1,1) and the radius = 5. N= 14 as there are 14 points of interest. However, only 9 out of 14 points fall inside the circle. In order to apply binary search on the points within the circle, we need to sort them and you will use merge sort to sort those points. The sorting criteria was discussed above. Let us see the sample input/output based on the given picture.



Sample data in Input file in.txt (the red points are outside of the circle):

```
-1  1  5  14
3  1
-6  -2
-4  2
4  -4
2  4
-1  3
2  2
0  -2
-4  -2
-6  6
4  4
-2  4
2  -2
-4  6
```

Sample output file out.txt:

```
-4  -2
-4  2
-2  4
-1  3
0  -2
2  -2
2  2
2  4
3  1
```

“filtered and sorted data written to out.txt” *//Note that you must close your file before going to search phase. Otherwise file will not be generated with your data until yo complete all the search process.*

```
Search input (x y): 2 -1
Output: Not found
Search input (x y): 3 1
Output: Found at record 9
Search input (x y): 2 -2
Output: Found at record 6
Search input (x y): -4 2
Output: Found at record 2
Search input (x y): -4 -2
Output: Found at record 1
Search input (x y): -999 10
Output: exit
```

Requirement:

You must have to use file IO, Merge sort and binary search for your solution. Without using them, you will get zero. The output must have to match with the sample output format. Do not add additional space, extra characters or words with the output. Next, you must have to write a well structure code. There will be a deduction of 2 points if the code is not well indented.

Your code should also have the following functions:

1. ReadData(): this function reads the required data from the in.txt file and return the data that are stored in your preferred data structure. Hints: this function might require dynamic memory allocation and return pointer after reading the data. Also, you might need to, pass references of one or more variables when you call the function so that the function can update those variables.

2. FilterData(): this function takes the data you received from ReaData() function and returns the filtered points that are inside or on the line of the circle. Hints: similar to the ReadData function, you might require to dynamically allocate memory and pass reference to appropriate variables to keep track how many points are inside the circle.

3. Standard MergeSort(), Merge(), BinarySearch() functions. You can take help from the uploaded code, but you should type in the code for them and modify them to fit the requirement and to fit your data structure.

4. SearchPhase(): This function takes your sorted data that was modified by the merge sort. Then it continuously prompts the user for search points and show the result or exit based on the provided criteria.

Additional Requirement:

Your program must have to compile in Eustis to get points. Note that you can use <math.h> library if you need. In that case you have to use –lm option while compiling your code.

For example: $*gcc filename.c -lm*

Rubric:

1) Filtering the points that are on the line or inside the circle: 2
2) Applying Merge sort properly and writing the results into out.txt file: 6.5
3) Performing Binary search properly and output: failing each test case -1.5 points. Total 4.5
4) There is no point for well structured, commented and well indented code. However, bad structure, and bad indented code will get -2 points penalty.

Please see the lecture slides and uploaded codes for learning merge sort and binary search and File I/O.

Remember, we had one lab on File I/O in the beginning of the semester. We have used ***fscanf*** and ***fprintf*** while reading and writing files, respectively. You can find example codes in c_review folder

You can take help from the codes that I have uploaded. However, they should be typed by you and modified as needed

For any clarification on the problem, don’t hesitate to contact us!

Deadline:
See the deadline in Webcourses.
The assignment will accept submissions up to 24 hours after the deadline with 20% late penalty. But it will not be a wise decision to delay your submission as you have quiz in the following week. Also, note that deadline means the last date of submission. It does not mean that you have to submit your assignment on that day. You can submit your assignment earlier too!