Whenever you are using double/float, calculation should be precise up to 0.0001
Use typedef, struct, enum and functions as much as possible to make your code readable and modular.

Q1. See the student_record.txtPreview the document file in Lab 8 folder.

The format of the file is as follows:

Every line has a student record in the following format

#Firstname Lastname BloodType BranchType Year Height_in_CM Weight_in_KG
#BloodType 0-ANEGATIVE,1-BNEGATIVE,2-ABNEGATIVE,3-ONEGATIVE, 4-APOSITIVE, 5- BPOSTIVE, 6- ABPOSITIVE, 7-OPOSITIVE
#BranchType 0-AI, 1-BM, 2-BT, 3-CHE, 4-CHY, 5-CSE, 6-EE, 7-ES, 8-MA, 9-ME, 10-MSME

#Year 1-4

(a) Write a program to read 1000 student records from stdin and print these records. Defining appropriate struct for student_record_t is mandatory.

While printing the record, print strings for BloodType and BranchType instead of their code.

For example,

Bruce Baner ANEGATIVE ES 3 180 70

The above indicates that the student's first name is Bruce, last name is Baner, Blood type is ANEGATIVE, Branch is ES, Year of study is 3, height is 180cm and weight is 70kg

Use file redirection to connect file to stdin as follows.

On Linux do the following:

```
./name_of_your_program < student_records.txt
```

On windows either follow the steps given in this link (https://stackoverflow.com/questions/10824178/how-do-you-redirect-standard-input-to-a-file-in-the-windows-command-line (Links to an external site.)) or just copy from the student_record.txt file and paste it in the windows command prompt/console.

(b) Print student records for highest and lowest student (in terms of height)

(c) Print student records for heaviest and lightest students (in terms of weight) for each branch and year. Total of 11 branch and 4 years are there so there would be 88 such records printed.

(d OPTIONAL) Assume the following rules. Any XNEGATIVE can donate to XPOSITIVE (where X is A, B, AB or O). ABPOLARITY can accept blood from any XPOLARITY where the polarities (-ve or +ve) match and X is (A, B, AB or O). OPOLARITY can donate blood to any XPOLARITY where polarities match and X is (A,B,AB, or O). Randomly select a student and find donors for this student in the same branch. Test this by writing appropriate function in main.

(e OPTIONAL) Find out for a randomly selected student, which student this selected student can donate to in the same branch.

Q2. Define a struct xypoint_t to represent points in x,y coordinate system and rtpoint_t to represent points in r,theta coordinate system. Use double for all coordinates.

(a) write a function that accepts xypoint_t as a parameter and returns rtpoint_t (the coordinates of the same point in r,theta system). Test this function by writing appropriate testing code in main()

(b) write a function that accepts rtpoint_t and returns corresponding xypoint_t. Test this function by writing appropriate testing code in main()

(c OPTIONAL) define a typedef triangle_t for array of 3 xypoint_t. write a function is_triangle() that accepts triangle_t and returns 1 if triangle inequality holds for all the three points, 0 if the inequality does not hold. Test this function by writing appropriate code in main(). Ensure that you also generate test cases for collinear points.