Code Explanation

Name: Prithvi Karthik
Roll Number: 2022101020

Q1. A C program that performs multiplication, subtraction, and addition of two complex numbers using structures in functions.

In this program, a struct has been defined with the name `Complex_Number` to define a complex number. It has two variables associated with it `real` and `imaginary` to denote the real and imaginary parts of a complex number.

For each of the operations, an individual function has been defined. In each of them, we define a new complex number `c` and then calculate its real and imaginary parts, before returning its value.

- Addition: For this, we add the real and imaginary components of the two complex numbers separately to get the final sum of the two numbers.
- Subtraction: To find the difference between two complex numbers, the real and imaginary components are subtracted separately to find the difference.
- Multiplication: As we do in regular arithmetic, we calculate the real part as the product of real components subtracted by the product of imaginary components of the two complex numbers. Similarly, the imaginary part is calculated to be the sum of the product of the real component of one number and the imaginary component of the other number.

The `main` function contains a while loop that continues to run until the value of a variable `flag` is set to 1. Inside the loop, the user is asked to select an operation to perform by entering a value (1 / 2 / 3 / 4). Using a switch statement, if the user inputs 1, the program prompts the user to enter two complex numbers and calls the add function to perform the addition. Similarly, if the user inputs 2, the program performs the subtraction of two complex numbers. And, if the user inputs 3, the program performs the multiplication of two complex numbers. If the user inputs 4, the flag is set to 1, causing the while loop to terminate and the program to end.

Q2. Write a structure to store the email, name, age (between 17 to 22), roll no., and hostel address of students (more than 20). Store the information of the students.

(a) Write a function to print the names of all the students aged 19.

(b) Create a function that prints the names of all students whose email addresses start with ’I’ or ’T’.

(c) Write another function to display the details of the student whose email is given (i.e. email entered by the user)

This program aims to store student details and perform functions on the input data. It consists of an array of structs to store information about the students including their name, age, roll number, email, and hostel address. Initially, user input is taken to find the number of students `n`. Using that input, a loop is run `n` times, and user input is taken about student data and stored in the struct.

As required, there are three functions defined:

- `print_students_with_age_19` iterates through the array of students, checking if their age is 19. If a student's age is 19, their name is printed out.
- `print_students_with_email_starting_with_I_or_T` iterates through the array of students, checking the first letter of their emails and checking if the character matches 'i'/'I'/'t'/'T' (since email Ids are case insensitive). If an email matches the criteria, the student's name and email are printed.
- 'student_email_available` checks whether a student's email matches a user inputted email. User input is asked and compared against the email Ids of all the students in the `students` array. If the email matches, the details of the student are printed.

Q3. Create a C program that capitalizes the first and last letters of every word in a file.

This program converts the first and last letters of every word of an input file into capital letters. It does this by reading each character of the file and grouping them into the 'words'. The program stops once the character reaches the end of the file (EOF).

Whenever the character `c` read from the file is a letter, it is added to the array `word` which is tracking the current word being read from the file. If the character read is a punctuation (",", ".", "!", "?", "&", "(", ")", ";", "-", ":"), a space or a newline, the `word` is considered to 'end'. Then the first and last letters of the word are converted to uppercase by subtracting 32 from their int (ASCII) value if they are lowercase. The output is printed to the terminal as saved in a new `temp.txt` file.

After converting the all words to uppercase, the text files are read and all its content is written to our source file `to_capitalise.txt`. After this, the temp file is deleted using the function `remove()`.

Q4. Write a C program to read and write student email, roll no, and marks of assignment-1,2,3 in a file.

This code reads, writes, and prints the data of multiple students from/to a binary file. There is a struct `student` defined that stores the email Ids, roll numbers, and assignment marks of the students.

The code opens the binary file `students.bin` in read mode to check if there are any previously saved data. If available, it allocates data for an array of students and iterates through it while printing out the student data. Then it asks the user to input whether the program should end or continue and overwrite the existing data. If there is no `students.bin` file, a message is printed and a new file is created.

The user is prompted to input student data. This data (number of students and student array) is written to the binary file using `fwrite` and is checked by reading the file back again using `fread` and the value is printed.