Question 1

You are required to implement an **Academy** class in C++ that manages a list of **students**. The academy can have different types of students, such as Bachelor students and Master students. You need to incorporate polymorphism to handle different types of students and provide methods to **add and display students.**

Please find the skeleton code of q6.cpp to find the code of the following classes. The main function is also given.

Student Base Class:

- Contains protected attributes **name** and **age**. (you are not allowed to add more members)
- Provides a constructor to initialize these attributes.

BachelorStudent Derived Class:

- Inherits from **Student**.
- Contains an additional attribute **major**. (you are not allowed to add more members)
- Provides a constructor to initialize all attributes.

MasterStudent Derived Class:

- Inherits from **Student**.
- Contains additional attributes **researchTopic** and **researchYears**. (you are not allowed to add more members)
- Provides a constructor to initialize all attributes.

Academy Class:

- **Student *students[4].** Assuming that we only have four students in academy.
- We are setting **number_of_students = 4** in a constructor, because there are only four students in our academy.
- **Studentcount** can be used as an iterator, it is initialized with 0. Now it's up to you how to use it but it is suggested to increment by one in a **addStudent()** method.

· Provides a constructor.

Task

1. Write a function **addStudent(/*you can add the arguments*/)** in Academy class to add the **Bachelor's** student in *students list.
2. Write a function **addStudent(/*you can add the arguments*/)** in Academy class to add the **Master** student in *students list. (Apply the concept of function overloading).
3. Write a function **displayStudents(/*you can add the arguments*/)** in Academy class to display the details of all the students in *students list. Look at the expected output.
4. You are also responsible for figuring out the methods need to be implemented in Student, BacherlorStudent or MasterStudent classes. Now it's up to you, no restriction on that.

Expected Output

Student Type: Bachelor
Name: Alice, Age: 20, Major: Computer Science
Student Type: Bachelor
Name: Charlie, Age: 21, Major: Mechanical Engineering
Student Type: Master
Name: Bob, Age: 25, Research Topic: Artificial Intelligence, Research Years = 2
Student Type: Master
Name: David, Age: 26, Research Topic: Data Science, Research Years = 1

Question 2

For the following question make a class Quiz and a nested Struct MCQ for the implementation.

In this question, your task is to implement the readfile(string) function to make a console based quiz application. The file questions.txt contains lines in the following format:

question: option1: option2: option3: option4: correct Option

Read the file, and prompt the user to give his answer for every question. If the answer is correct the score will be incremented by 1. If the answer is incorrect the system will display "wrong, correct option is …". Read all questions from the file, get user's answer, match it with correct answer and calculate the score. At the end score must be returned.

Note: There are some extra strings for example: option1, option2, question etc. It is mandatory to write these strings. In short, your output must be formatted 100% like the **given sample output** for correctness. The text file i.e. questions.txt should remain untouched, any modification in file will result in straight zero.

Question 3

Make a struct student (name, roll number & age). Make an object student1 in main(). Write it into a binary file & then read it into student s2.