Question 1: (5 marks)

Write a program in Java using **linked list** to manage information about persons. Variables used to store information about a person are:

- name - the name of a person (character string).
- age - the age of a person (integer value).

You should write the MyList class, which is a **linked list** data structure to store person information. The following functions should be included in the MyList class:

- void addLast(String xName, int xAge) - check if **the first letter of xName is 'B'** (i.e. xName.charAt(0) == 'B') then **do nothing**, otherwise add new person with name=xName, age=xAge to the end of the list.
- void addFirst(String xName, int xAge) - check if **the first letter of xName is 'B'** then **do nothing**, otherwise add new person with name=xName, age=xAge to the begining of the list.
- void addMany(String [] a, int [] b) - ***this function is given***.
- void ftraverse(RandomAccessFile f) throws Exception - display all nodes in the file f in format: (name, age). ***This function is given***.

- void f1() – Test addLast function. You do not need to edit this function. Your task is to complete the function addLast(String xName, int xAge) function only.
  With the given data, the content of **f1.txt must be the following:**
  (A0,9) (A7,13) (A5,7) (A3,11) (A4,9) (A2,12) (A6,5) (A1,6)

- void f2() – Test addFirst function. You do not need to edit this function. Your task is to complete the function addFirst(String xName, int xAge) function only.
  With the given data, the content of **f2.txt must be the following:**
  (A1,6) (A6,5) (A2,12) (A4,9) (A3,11) (A5,7) (A7,13) (A0,9)

- void f3() – There is a given MyList object h. Using addLast method to add to h all elements having age>4.
  With the given data, the content of **f3.txt must be the following:**
  (C4,9) (C6,3) (C3,7) (C6,1) (C7,9) (C5,2) (C1,5) (C2,6)
  (C4,9) (C3,7) (C7,9) (C1,5) (C2,6)

- void f4() - delete the first node having age<6.
  With the given data, the content of **f4.txt must be the following:**
  (C4,9) (C6,3) (C3,7) (C6,1) (C7,9) (C5,2) (C1,5) (C2,6)
  (C4,9) (C3,7) (C6,1) (C7,9) (C5,2) (C1,5) (C2,6)

- void f5() – sort the list ascendingly by name.
  With the given data, the content of **f5.txt must be the following:**
  (C4,9) (C6,3) (C3,7) (C6,1) (C7,9) (C5,2) (C1,5) (C2,6)
  (C1,5) (C2,6) (C3,7) (C4,9) (C5,2) (C6,1) (C6,3) (C7,9)

- void f6() – reverse the list.
  With the given data, the content of the file **f6.txt must be the following**::
  (C4,9) (C6,3) (C3,7) (C6,1) (C7,9) (C5,2) (C1,5) (C2,6)
  (C2,6) (C1,5) (C5,2) (C7,9) (C6,1) (C3,7) (C6,3) (C4,9)

- void f7() – append another list to the list.
  With the given data, the content of the file **f7.txt must be the following**::
  (C4,9) (C6,3) (C3,7) (C6,1) (C7,9) (C5,2) (C1,5) (C2,6)
  (C4,9) (C6,3) (C3,7) (C6,1) (C7,9) (C5,2) (C1,5) (C2,6) (D4,19) (D6,13) (D3,17)

- void f8() – delete the position i = 3 (The first position is 0).
  With the given data, the content of the file **f8.txt must be the following**::

(C4,9) (C6,3) (C3,7) (C6,1) (C7,9) (C5,2) (C1,5) (C2,6)
(C4,9) (C6,3) (C3,7) (C7,9) (C5,2) (C1,5) (C2,6)

- void f9() – Change the name of the first node having name = “C6” to “CX”.
  With the given data, the content of the file **f9.txt must be the following**::
  (C4,9) (**C6**,3) (C3,7) (C6,1) (C7,9) (C5,2) (C1,5) (C2,6)
  (C4,9) (**CX**,3) (C3,7) (C6,1) (C7,9) (C5,2) (C1,5) (C2,6)