**Question 1: (4 marks)**

(Do not pay attention to real meaning of objects, variables and their values in the questions below).

In this question you should complete some methods in **MyList.java** file.

The class Car with 2 data members: owner and price is given and you do not need to edit it. The MyList class is a linked list of Car objects. The following methods should be completed:

* void addHead(String xOwner, int xPrice) - check if the last character of xOwner = 'B' or xPrice>100 then **do nothing**, otherwise add new car with owner=xOwner, price=xPrice to the head of the list. (price can get arbitrary value, even negative).
* void **f1()** – This method is used to test the addHead method above. You do not need to edit this function. Output in the file **f1.txt** must be the following:

(AF,4) (AE,6) (AD,2) (AC,7) (AA,9)

* void **f2()** – There is a given objects x. You should write statements so that x will be insert before the first element which has price is greater than 10. Output in the file **f2.txt** must be the following:

(AF,4) (AE,6) (AD,12) (AC,7) (AA,9)

(AF,4) (AE,6) (X,1) (AD,12) (AC,7) (AA,9)

* void **f3()** – Suppose the list contains at least 3 elements. Delete the Node after first node having price>10 and the second character of owner equal ‘C’. Output in the file **f3.txt** must be the following:

(AE,19) (AC,20) (AF,4) (AE,6) (AD,12) (AC,7) (AA,9)

(AE,19) (AF,4) (AE,6) (AD,12) (AC,7) (AA,9)

* void **f4()** – Sort the list ascendingly by owner. Output in the file **f4.txt** must be the following:

(AE,19) (AC,20) (AF,4) (EE,6) (AD,12) (BC,8) (AA,9)

(AA,9) (AC,20) (AD,12) (AE,19) (AF,4) (BC,8) (EE,6)

**Question 2: (4 marks)**

In this question you should complete some methods in **BSTree.java** files.

The class Car with 2 data members: owner and price is given and you do not need to edit it. The BSTree class is a binary search tree of Car objects. The variable **price is the key of the tree**. The following methods should be completed:

* void insert(string xOwner, int xPrice) - check if the second character of xOwner equals 'B' or xPrice is even number then **do nothing**, otherwise insert new car with owner=xOwner, price=xPrice to the tree.
* void **f1()** – You do not need to edit this function. Your task is to complete the insert(...) function above only. Output in the file **f1.txt** must be the following:

(BA,5) (A2,3) (A1,1) (BD,61) (FA,47) (G1,35) (B2,25) (Y5A,105)

(A1,1) (A2,3) (BA,5) (B2,25) (G1,35) (FA,47) (BD,61) (Y5A,105)

* void **f2()** – Perform post-order traversal from the root but display to file f2.txt nodes having price in the interval [3,50] only. **Hint:** Copy the function preOrder(...) to preOrder2(...) and modify it. Output in the file **f2.txt** must be the following:

(BA,5) (A2,3) (A1,1) (BD,61) (FA,47) (G1,35) (B2,25) (Y5A,105)

(BA,5) (A2,3) (FA,47) (G1,35) (B2,25)

* void **f3()** – Perform preOrder traversal from the root and delete by copying the first node having both 2 sons and price in the interval [30,70] Output in the file **f3.txt** must be the following:

(A1,1) (A2,3) (BA,5) (B2,25) (G1,35) (FA,43) (X2,47) (BD,61) (Y5A,105)

(A1,1) (A2,3) (BA,5) (B2,25) (G1,35) (X2,47) (BD,61) (Y5A,105)

* void **f4()** – Perform inOrder traversal from the root and find the first node p having right son and price in the interval [30,70]. Rotate p to left about its’ right son. Output in the file **f4.txt** must be the following:

(BA,5) (A2,3) (BD,61) (A1,1) (FA,43) (Y5A,105) (G1,35) (X2,47) (B2,25)

(BA,5) (A2,3) (A1,1) (Y5A,105) (BD,61) (FA,43) (G1,35) (B2,25) (X2,47)

**Question 3: (2 marks)**

In this question you should complete some methods in **Graph.java** file.

The class Graph is the implementation of a graph. The following methods should be completed:

* void **f1()** - Perfom depth-first traversal (to the file f1.xt) from the vertex i=1 (the vertex B) but display vertices with their deegrees in bracket. **Hint**: copy depth(...) to depth2(...) and modify the latter one. Content of the output file **f1.txt** must be:

B G A E F I C H D

B(1) G(2) A(4) E(3) F(3) I(3) C(1) H(2) D(1)

* void **f2()** – Apply the Dijkstra’s shortest path algorithm to find the shortest path from the vertex 0 (A) to the vertex 4 (E). (Note that in the weighted matrix, the value 999 is considered as infinity). Write 2 lines into the file f6.txt. The first line contains the list of vertices in the shortest path. The second lines contains shortest distances to the vertices in the first line. Content of the output file **f2.txt** must be:

A C F E

0 9 11 20

* void **f3()** – Supposed the given graph has Euler's cycle. Apply the pseudocode in the Graph.java file to write statements to find the Euler's cycle from the vertex 1 (B). Output in the file **f3.txt** must be the following:

B D E D C B E G F A B