****
WRITE YOUR EXAM NUMBER HERE:

****
IMPORTANT READ CAREFULLY
****

1- There are nine questions in this paper. Answer ALL NINE questions.
2- Edit your code in java files in the BlueJ project, NOT in this file
3- SAVE your work regularly. It is your responsibility to save it.
4- DO NOT use JOptionPane in the exam.
5- DO NOT change any of the code apart from what is specified in a question. Failing to comply may result in heavy penalties.
6- DO NOT IMPORT additional packages.
7- NEVER change method names and the type and order of parameters specified in the question: if the question asks to write a method with signature

    public void Foo(int n, double d)

then writing: public void foo(int n, double d) will be wrong (f should be capital) and public void Foo(double d, int n) will also be wrong (parameter int should be before parameter double)
8- Check carefully that the output of your program is EXACTLY the one required by the exercises. For example if asked to print a string s (e.g. "Wrong argument"), print exactly that string i.e. use System.out.println("Wrong argument"), do not modify the output string.
9- NEVER write two classes in the same java file.
10- If a question provides some code to test your code, to pass those tests is not a guarantee that you will get marks; the tests usually only guarantee that you have made no syntax errors
11- The Appendix contains help for some methods that can be used in the exam. Notice that not all of these methods will be needed to answer the questions. If available on the machine you are also allowed to use the java api documentation
12- ONLY use System.exit(0) if the question says so. Never use System.exit with any other argument e.g. NEVER use System.exit(-1)

****
Question 1 [6 Marks]
****

In the following class some of the instance variables are wrongly declared:
Modify only the types in declarations so that the class will compile
(do not modify the name of the variables or their initial values):

```
public class Ex1 {

private Account y='h';     //modify only the type
private char[] x="hello";      //modify only the type
private float c =false;    //modify only the type
```

```
private float s ="double";       //modify only the type

   public void getValuesA(){
   System.out.println(x+" "+y+" ");
   }

   public void getValuesB(){
   System.out.println(s+" "+c);
   }

}
```

****
Question 2 [6 Marks]
****

Insert in the following class a constructor that will initialise the instance variables name and status.
The constructor has two arguments: a string personName and an integer age; name is initialized by personName.
About the instance variables status: if age is over 18 the constructor initializes the instance variable status to "can vote"
otherwise the constructor initializes the instance variable status to "cannot vote".
For example
the statement (new Ex2("John",40)).getValues() will return "John can vote"
the statement (new Ex2("John",15)).getValues() will return "John cannot vote"

```
public class Ex2 {

private String name;
private String status;


   public String getValues(){
      return ""+name+status;
   }
   public static void main(String arg[]){ // you can use the main method to test your code
      System.out.println((new Ex2("John",40)).getValues());
   }

}
```

****
Question 3 [9 Marks]
****

Consider the following Accumulator class with missing code for the method

'getOverM(int m)'.
getOverM should return the sum of the elements of the array A if the value of such sum is greater than m and should return -1 if the sum of the value of elements of the array A is less or equal to m.

For example if A is the array {2,4,3,5,8} then
getOverM(3) will return 22
getOverM(2) will return 22
getOverM(7) will return 22
getOverM(30) will return -1

(Hint: the length of an array A is given by A.length)

Insert the code for the body of the method getOverM.

```
public class Accumulator {
   private int[] A;

   public Accumulator(int[] X) {
      A= new int[X.length];
      for (int i=0; i<X.length; i++)
         A[i] = X[i];
   }

   public int getOverM(int m) {

   }

    public static void main(String args[]){ // you can use the main method to test your
code

     int[]  A = {2,4,3,5,8};

     int r=new Accumulator(A).getOverM(3); //change argument to test different cases
     System.out.println(r);
   }

}
```

****
Question 4 [7 Marks]
****

Fill in the class method main of class Ex4 with code that
create an array of size 3 containing strings and fills it with command line arguments.
The user should provide at least one command line argument.
If the user provides only one command line arguments then that element of the

array should be printed.
If the user provides two or three command line argument then the first and second element of the
array should be printed.
However if the user provides more than three command line arguments the program should print
"Too many arguments".
(You can test for the number of arguments or catch an ArrayIndexOutOfBoundsException for this purpose)

For example
java Ex4 A will print the string "A"
java Ex4 A B C will print the string "A B"
java Ex4 A B will print the string "A B"
java Ex4 A B C D will print the string "Too many arguments"

```
public class Ex4 {

  public static void main(String[] args) {

  }
}
```

****
Question 5 [12 Marks]
****

Consider the following class:

```
public class Message
{
   private String sender;
   private String receiver;
   private String content;

   public Message(String ssender, String rreceiver, String ccontent){
      sender=ssender;
      receiver=rreceiver;
      content=ccontent;
    }

   public String getSender(){
      return sender;
    }

    public String getReceiver(){
      return receiver;
    }
```

```
    public String getContent(){
      return content;
    }
}
```

Write a subclass of the Message class called NationalMessage

The constructor of NationalMessage has signature
  public NationalMessage(String ssender, String rreceiver, String ccontent)
this constructor will call the superclass constructor on its arguments

The class NationalMessage has a private instance variable of boolean type called isNational.

This variable is initialised in the constructor NationalMessage as follows:

if the strings ssender and rreceiver both starts with the string "UK" then isNational is set to true, otherwise isNational is set to false.

The class NationalMessage also has a getter for the variable isNational with signature
 public boolean getIsNational()

Hence for example for an instance of NationalMessage created with
new NationalMessage("UKJohn","UKMark","aa")
getIsNational() would return true

For an instance of NationalMessage created with
new NationalMessage("USAJohn","FRMark","aa")
getIsNational() would return false

(hint: you can use the method startsWith(String prefix) to check if a string starts with the substring prefix)

****
Question 6 [13 Marks]
****

In the following program, the code in the method countNational is missing.
Write the code for this method, which takes as argument the ArrayList of Messages messageList and
returns the number of messages in messageList which are of type NationalMessage and for which getIsNational() is true

The method should also:

- handle the the case of the ArrayList messageList being null. In case messageList is null countNational
should print "Error" and return -1

(Hint:
1- use m.size() to get the number of elements in the ArrayList m
2- use the keyword instanceof to check if an object is an instance of a class)

For example,
-if messageList contains only two NationalMessage objects, in the first isNational is false and in the second isNational is true then countNational will return 1
-if messageList contains 10 instances of Message and 5 of these objects are instances of NationalMessage and in all instances of NationalMessage isNational is true then countNational will return 5
-if messageList is null countNational will print the string "Error"

```
import java.util.*;

public class Ex6 {

    public static int countNational (ArrayList<Message> messageList) {

    }
 public static void main(String[] args){ // you can use this main method to test your code
        ArrayList<Message> messageList = new ArrayList<Message>();
        messageList.add(new NationalMessage("UKJohn","UKMark","aa"));
        messageList.add(new NationalMessage("UKJohn","FRJean","aa"));
        messageList.add(new Message("Mike","John","aa"));
        System.out.println(countNational(messageList));
    }
}
```

****
Question 7 [15 Marks]
****

In the class Ex7 write the code for the method nationalMessage(String fName)

The method must:
- first open the file fName and
- then look in each line in the file if the line begins with two strings both starting with "UK" and in that case create an object NationalMessage with arguments the first string in that line, the second string in that line and the third string in that line; after that it stops reading the file and returns that object.

However,

- if an IOException occurs, the method should print "IO Error" and return null

To test your program, use the file "data.txt" (available in your current directory), each line of which contains three strings,

As an example, if "data.txt" is
EU CH nosecret
UKa UKb meeting
UKa UKb 300
then
nationalMessage("data.txt") will return an instance of NationalMessage created with
NationalMessage("UKa","UKb","meeting")

(Hint:
1) the statement: String[] lines=line.split(" ");
will put in the array lines the word in the string line separated
by an empty space, e.g. if line is
"John 1000 2000" then after String[] lines=line.split(" ");
lines[0] is "John", lines[1] is "1000" and lines[2] is "2000")
2) you can use the method startsWith(String prefix) to check if a string starts with the substring prefix)

```
import java.io.*;

public class Ex7 {

    public static NationalMessage nationalMessage(String fName){

    }

}
```

****
Question 8 [12 Marks]
****

Add to the class Ex8 (below) the code for the following method

```
    public static String fromSender(ArrayList<Message> messageList, String pname)
```

which takes as arguments an ArrayList of Message objects (as given in question 5) named messageList, a String pname and returns a string containing the content of all messages in messageList that had been sent by the user pname

For example, if the ArrayList messageList contains three messages with sender "UKMark", the first message has content "my first message ", the second has content "my second message ",
the third has content "my third message ", then

fromSender(messageList, "UKMark")

will return the string
"my first message my second message my third message "

```
import java.util.*;
 public class Ex8 {

    public static String fromSender(ArrayList<Message> messageList, String pname){


    }
  public static void main(String[] args){ //you can use this main method to test your
code

        ArrayList<Message> messageList= new ArrayList<Message>();
        messageList.add(new Message("UKMark","UKJohn","message1 "));
        messageList.add(new Message("John","Don","ah"));
        messageList.add(new Message("UKMark","UKJohn","message2 "));
        System.out.print(fromSender(messageList, "UKMark") );
    }
}
```

****
Question 9 [20 Marks]
****

Consider the following interface

```
public interface SecureMessage{
 public int[] encrypt(String content,int key);
 public String decrypt(int[] content,int key);
}
```

a)
Write a subclass of Message called SecureNationalMessage implementing the interface SecureMessage.
The constructor SecureNationalMessage has signature
public SecureNationalMessage(String ssender, String rreceiver, String ccontent, int key)

overall SecureNationalMessage is the same as NationalMessage but has getters for the instance variable key and also in SecureNationalMessage the methods encrypt and decrypt are implemented.

The method encrypt first converts the string content into an array of characters (hint you can use the method toCharArray() to convert a string into an array of characters), then convert each character

c in the array in the integer (int)c and finally adds to this integer the integer key

for example if the key is 5 then the character 'b' is mapped to 98+5=103

for example encrypt("any word",5) should return the array
102,115,126,37,124,116,119,105

The method decrypt takes the array of integers content as input and builds the return string by subtracting the integer key from each integer in the array, casting the resulting integer into a char and concatenating that char to the string.

you should verify that decrypt(encrypt("any word",5),5)="any word",
i.e. decrypt is the inverse of encrypt for strings which contain letters of the alphabet

(this part is worth 12 marks)

b)
Write a class called ListSecureNationalMessages with a method with signature

```
public static void secureNational (ArrayList<Message> messageList,int key)
```

this method takes an ArrayList of Message objects and prints the encrypted content of messages of the elements in messageList which have type SecureNationalMessage.

For example if the ArrayList test is defined as follows

```
ArrayList<Message> test=new ArrayList<Message>();
test.add(new SecureNationalMessage("a","b","aa",5));
test.add(new SecureNationalMessage("a","b","bb",5));
test.add(new SecureNationalMessage("a","b","cc",5));
```

then secureNational(test,5) will print
102102103103104104

(this part is worth 8 marks)

********** APPENIDX **********
********** DOCUMENTATION **********
(you may need only some of these methods)
****
SOME USEFUL ARRAYLIST METHODS
boolean add(E e)
Appends the specified element to the end of this list.
void add(int index, E element)
Inserts the specified element at the specified position in this list.

get(int index)
Returns the element at the specified position in this list.

size()
Returns the number of components in this list.

****

To check if an object is an instance of a class c use the operator instanceof
o instanceof c will return true if o is an instance of c and false otherwise.

****

Integer.parseInt(String s)
Parses the string argument as an integer. Throws NumberFormatException
if the string does not contain a parsable integer.

****

% is the Remainder operator. (n % m) returns the remainder of the division
of n by m, e.g. 3 % 2 = 1

****

To generate a double random number between 0 and 1 you can use
Math.random()

****

To get an integer number that shows the number of elements in an array a,
you can use a.length

****

SOME FILE READING CONSTRUCTORS AND METHODS

FileReader(String fileName)
Creates a new FileReader, given the name of the file to read from.

BufferedReader(Reader in)
Create a buffering character-input stream that uses a default-sized input buffer.

String readLine()
Read a line of text.

to read a file until its last line you should check if the line is null, e.g. if the current line
of the file is in the String line then you can use

while(line != null) ...

****

SOME USEFUL STRING METHODS

char charAt(int index)
Returns the character at the specified index.

boolean equals(Object anObject)
Compares this string to the specified object.

int indexOf(int ch)
Returns the index within this string of the first occurrence of the specified character.

int lastIndexOf(String str)
Returns the index within this string of the rightmost occurrence of the specified substring.

int length()
Returns the length of this string.

String substring(int beginIndex, int endIndex)
Returns a new string that is a substring of this string.

String trim()
Returns a copy of the string, with leading and trailing whitespace omitted