Question 1: (12 Marks)

a- what is the output made by the following portion of code

	Answer:

`int i =1, V=0, W=0;`
`while (i<=10 && V-W<=10) {`
`   W = V ;`
`   V = 2* V + i;`
`   i++;`
`   System.out.println("V ="+V);`
`}`
	V =1
	V =4
	V =11
	V =26

b- Suppose that we have an array **`Marks`** that has the length 5 and the following content

Marks =

10	20	15	17	9

- **What is the output of the following code segment:**

	Answer:

`   int s =0 ;`
`   for(int i=0;i<Marks.length-1;i++)`
`     if (Marks[i]< Marks[i+1]){`
`      System.out.println("a["+i+"]="+`
`Marks[i]);`
`      s = s + Marks[i];`
`     }`
`   System.out.println("s = " + s);`
	a[0]=10
	a[2]=15
	s = 25

- **What is the output of the following code segment:**

	Answer:

`int i = 0, s = 0, c = 0;`
`while (2-c>0){`
`   if (Marks[i]>15) {`
`      c++;`
`      System.out.println("c = “+c);`
`   }`
`   i++;`
`   s += c;`
`   }`
`System.out.println("result="+Marks[i]+" s="+s);`
	c = 1
	c = 2
	result=9 s=4

Question 2: `Given the following class Bicycle` (12 marks)

```
    public class Bicycle {
            private int gear;
            private int speed;
            public Bicycle(int startSpeed, int startGear) {
                    gear = startGear;
                    speed = startSpeed;
            }
            public int getGear() {
                    return gear;
            }
            public void setGear(int newValue) {
                    gear = newValue;
            }
            public int getSpeed() {
                    return speed;
            }
            public void applyBrake(int decrement) {
                    speed -= decrement;
            }
            public void speedUp(int increment) {
                    speed += increment;
            }
            public boolean equal(Bicycle B) {
               return (gear == B.gear && speed == B.speed);
            }
    }
```

complete the needed statements in the Main class according to the following :

```
public class Main {
   public static void main(String[] args){
   //1.     Declare and create two objects of type Bicycle:
   //   - C1 has an initial gear of 4 and an initial speed of 20,
   //   - C2 has an initial gear of 2 and an initial speed of 15.
      Bicycle c1 = new Bicycle(20,4);
      Bicycle c2 = new Bicycle(15,2);
```

```
//2.      Write the statement(s) to speedup object C2 with 5 units.
      c2.speedUp(5);
```

```
//3.      Write the statements to copy C2 into C1.

         int c2Speed = c2.getSpeed();
         int c2Gear= c2.getGear();
         c1 = new Bicycle(c2Speed,c2Gear);

// or
// int gear = c2.getGear();
// int speed = c2.getSpeed();
// int oldSpeed = c1.getSpeed();
// c1. applyBrake(oldSpeed);
// c1.setGear(gear);
// c1.speedup(speed);


// or
// if (c2.getSpeed() > c1.getSpeed() )
//  c1.speedUp(c2.getSpeed() - c1.getSpeed()  )
// else
//   c1. applyBrake( c1.getSpeed() - c2.getSpeed()   )
```

```
//4.    Write the statement(s) that displays 'YES' if C1 and C2 are equal.
        if ( c1.equal(c2) == true )
```

```
                    System.out.print("yes");








  }
}
```

Question 3 : (16 Marks)

a- Complete the methods of the following class (11 Marks)

```
public class MobileStore {

    private int[]    codes ;   // stores the code of the Mobile
    private double[] prices ; // stores the price of the Mobile
    private int[]    quantities ;// stores the quantity of Mobile
    private int counter ;   // counts the number of the inserted Mobiles

    public MobileStore (int size) {
         codes     = new int[size];
         prices    = new double[size];
         quantities = new int[size];
         counter = 0 ;
    }

    public double getMobilePrice(int index) {
    //This method returns the price of the Mobile located at index

     if ( index < 0 || index >= counter )
             return -1;
     else
             return prices[index];

   }

   public void insertMobile(int id, double pce, int qtity) {
  // this  method adds, if possible, the given data
  //(id, pce and qtity) related to a new mobile

   if ( counter >= prices.length )
                System.out.print("NoSpace");
        else
        {
                codes[counter] = id;
                prices[counter] = pce;
                quantities[counter] = qtity;
                counter++;
        }
  }


 public int cheapestMobile() {
   //returns the code of the available Mobile that has the minimum
// price. (available means that the quantity >0)


     int cheapestCode = -1;
     double min = prices[0];
     for ( int i = 0 ; i < counter ; i++){
             if ( prices[i] <= min && quantities[i] > 0){
                     min = prices[i];
                     cheapestCode = codes[i] ;
             }
     }
     return cheapestCode;


 }

  public boolean isMobileAvailable(int cd) {
  // if a Mobile with code cd exists and its quantity is greater than
  // zero then return true, and otherwise return false.

      for ( int i = 0 ; i < counter ; i++){
                  if ( codes[i] == cd && quantities[i] > 0)
                          return true;
      }
      return false;
  }



public void addQuantity(int cd, int newQuantity) {
 // newQuantity is added to the quantity of the Mobile
 // that has the code cd  (newquantity MUST be a positive number)


    if (newQuantity > 0 ){
      for ( int i = 0 ; i < counter ; i++){
                  if ( codes[i] == cd )
                          quantities[i]+=newQuantity;

      }


}
```

b- Give the UML representation of the class **MobileStore** **(5 Marks)**

```
                 MobileStore
----------------------------------------------------
   - codes: int []
   - prices: double[]
   - quantities: int[]
   - counter : int
----------------------------------------------------
   + MobileStore(size:int)
   + getMobilePrice (index:int):double
   + insertMobile(id:int, pce:double, qtity:int):void
   +  cheapestMobile ():int
   + isMobileAvailable(cd:int):boolean
   + addQuantity(cd:int, newQuantity:int):void
```