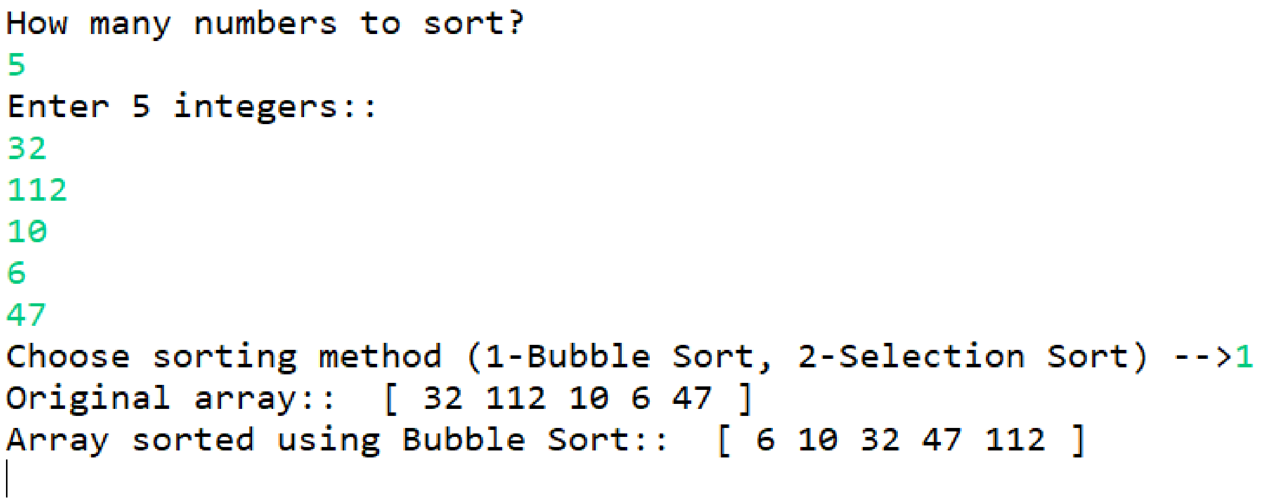
1. Write a program to take an integer array from the user and give the user a choice to sort using bubble sort (or) selection sort. Sort the array elements according to the selected algorithm of the user and display the sorted array.

```
import java.util.Scanner;

public class Sba4_01 {

	// static method to print array
	static void printarr(String s, int[] arr) {
		System.out.print(s + " [ ");
		for (int i : arr) {
			System.out.print(i + " ");
		}
		System.out.println("]");
	}

	// static method for bubble sort
	static int[] bubblesort(int[] arr) {
		int len = arr.length;
		for (int i = 0; i < len - 1; i++) {
			for (int j = 0; j < len - i - 1; j++) {
				if (arr[j] > arr[j + 1]) {
					int temp = arr[j];
					arr[j] = arr[j + 1];
					arr[j + 1] = temp;
				}
			}
		}
		return arr;
	}

	// static method for selection sort
	static int[] selectionsort(int[] arr) {
		int len = arr.length, temp = 0;
		for (int i = 0; i < len - 1; i++) {
			for (int j = i + 1; j < len; j++) {
				if (arr[j] < arr[i]) {
					temp = arr[i];
					arr[i] = arr[j];
					arr[j] = temp;
				}
			}
		}
		return arr;
	}
	public static void main(String[] args) {
		Scanner sc = new Scanner(System.in);
		System.out.println("How many numbers to sort?");
		int n = sc.nextInt();
		int[] numArr = new int[n];
		System.out.println("Enter " + n + " integers::");
		for (int i = 0; i < n; i++) {
			numArr[i] = sc.nextInt();
		}
		System.out.print("Choose sorting method (1-Bubble Sort, 2-Selection Sort) -->");
		int choice = sc.nextInt();
		printarr("Original array:: ", numArr);
		switch (choice) {
		case 1:
			printarr("Array sorted using Bubble Sort:: ", bubblesort(numArr));
			break;
		case 2:
			printarr("Array sorted using Selection sort:: ", selectionsort(numArr));
			break;
		default:
			System.out.println("INVALID CHOICE");
		}
		sc.close();

	}
}
```

Output :

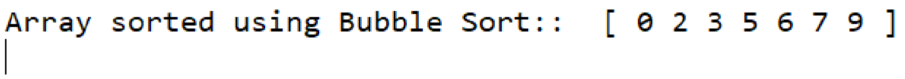


```
How many numbers to sort?
5
Enter 5 integers::
32
112
10
6
47
Choose sorting method (1-Bubble Sort, 2-Selection Sort) -->1
Original array::  [ 32 112 10 6 47 ]
Array sorted using Bubble Sort::  [ 6 10 32 47 112 ]
```

2. Write a program to implement insertion sort.

```
public class Sba4_02 {

    static int[] insertionsort(int[] arr) {
        int n = arr.length;
        for (int j = 1; j < n; j++) {
            int key = arr[j];
            int i = j - 1;
            while ((i > -1) && (arr[i] > key)) {
                arr[i + 1] = arr[i];
                i--;
            }
            arr[i + 1] = key;
        }
        return arr;
    }

    // static method to print array
    static void printarr(String s, int[] arr) {
        System.out.print(s + " [ ");
        for (int i : arr) {
            System.out.print(i + " ");
        }
        System.out.println("]");
    }

    public static void main(String[] args) {
        int[] numArr = { 5, 9, 7, 3, 6, 0, 2 };
        printarr("Array sorted using Bubble Sort:: ", insertionsort(numArr));
    }

}
```

Output :

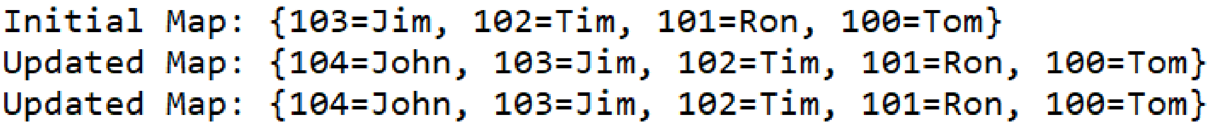


```
Array sorted using Bubble Sort::  [ 0 2 3 5 6 7 9 ]
```

3. Write a program to implement Hashtable and add at least 4 values into it, implement the putIfAbsent() method.

```
import java.util.Hashtable;

public class Sba4_03 {
        public static void main(String args[]) {
                Hashtable<Integer, String> m = new Hashtable<Integer, String>();
                m.put(100, "Tom");
                m.put(102, "Tim");
                m.put(101, "Ron");
                m.put(103, "Jim");
                System.out.println("Initial Map: " + m);
                // Inserts, as the specified pair is unique
                m.putIfAbsent(104, "John");
                System.out.println("Updated Map: " + m);
                // Returns the current value, as the specified pair already exist
                m.putIfAbsent(101, "Ron");
                System.out.println("Updated Map: " + m);
        }
}
```

Output :



```
Initial Map: {103=Jim, 102=Tim, 101=Ron, 100=Tom}
Updated Map: {104=John, 103=Jim, 102=Tim, 101=Ron, 100=Tom}
Updated Map: {104=John, 103=Jim, 102=Tim, 101=Ron, 100=Tom}
```

4. Create a class of Books with attributes:

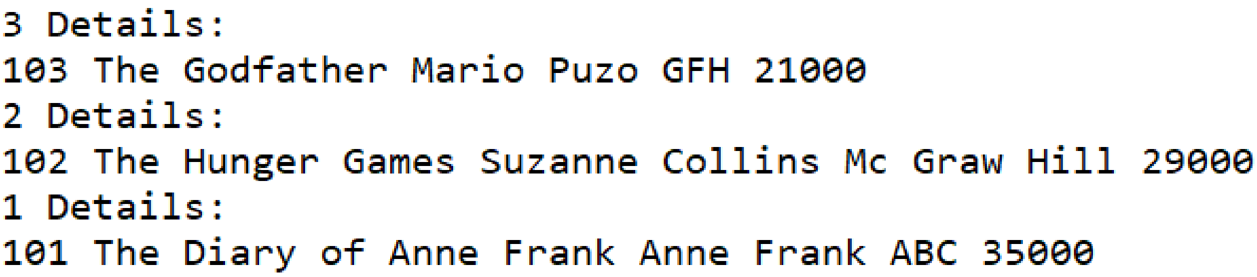
a) id
b) name
c) author
d) publisher
e) quantity sold

Implement a Hashtable to implement the objects of Books type. Print all the details of books by traversing through the Hashtable.

```
import java.util.Hashtable;
import java.util.Map;

class Book {
    int id;
    String name, author, publisher;
    int quantity_sold;

    public Book(int id, String name, String author, String publisher, int quantity_sold) {
        this.id = id;
        this.name = name;
        this.author = author;
        this.publisher = publisher;
        this.quantity_sold = quantity_sold;
    }
}


public class Sba4_04 {
    public static void main(String[] args) {
        // Creating map of Books
        Hashtable<Integer, Book> map = new Hashtable<Integer, Book>();
        // Creating Books
        Book b1 = new Book(101, "The Diary of Anne Frank", "Anne Frank", "ABC", 35000);
        Book b2 = new Book(102, "The Hunger Games", "Suzanne Collins", "Mc Graw Hill",
29000);
        Book b3 = new Book(103, "The Godfather", "Mario Puzo", "GFH", 21000);
        // Adding Books to map
        map.put(1, b1);
        map.put(2, b2);
        map.put(3, b3);
        // Traversing map
        for (Map.Entry<Integer, Book> z : map.entrySet()) {
            int key = z.getKey(); // key=3
            Book b = z.getValue(); // b=b3
```

```
                              System.out.println(key + " Details:");
                              System.out.println(b.id + " " + b.name + " " + b.author + " " + b.publisher + " "
+ b.quantity_sold);
                    }
          }
}
```

Output :



```
3 Details:
103 The Godfather Mario Puzo GFH 21000
2 Details:
102 The Hunger Games Suzanne Collins Mc Graw Hill 29000
1 Details:
101 The Diary of Anne Frank Anne Frank ABC 35000
```