

QUESTION 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.

package Assignment;

import java.util.Scanner;

public class BubbleSortORSelectionSort {

void bubbleSort(**int** arr[])

 {

int n = arr.length; //n=6

for (**int** i = 0; i < n-1; i++)

for (**int** j = 0; j < n-i-1; j++)

 {

if (arr[j] > arr[j+1])

 {

 // swap arr[j+1] and arr[j]

int temp = arr[j];

 arr[j] = arr[j+1];

 arr[j+1] = temp;

 }

 }

 }

void printArray(**int** arr[])

 {

int n = arr.length;

for (**int** i=0; i<n; ++i)

 System.out.print(arr[i] + " ");

 System.out.println();

 }

void sort(**int** arr[])

 {

 //1,2,3,4,6,9

int n = arr.length; //6

 // One by one move boundary of unsorted subarray

for (**int** i = 0; i < n-1; i++)

 {

 // Find the minimum element in unsorted array

int min_idx = i;//

for (**int** j = i+1; j < n; j++)

 {

if (arr[min_idx] > arr[j])

 min_idx = j;//5

 } // Swap the found minimum element with the first

int temp = arr[min_idx];

 arr[min_idx] = arr[i];

 arr[i] = temp;

 }

```

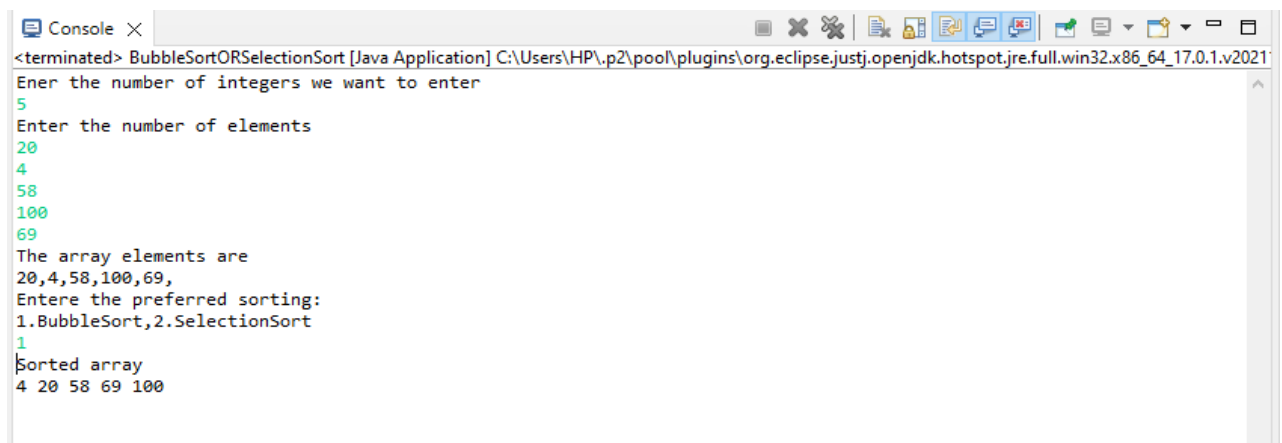
    }

    public static void main(String[] args) {
        System.out.println("Enter the number of integers we want to enter ");
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int[] arr = new int[n];
        System.out.println("Enter the number of elements");
        for (int i = 0; i < n; i++) {
            arr[i] = sc.nextInt();
        }
        System.out.println("The array elements are");
        for (int i = 0; i < n; i++) {
            System.out.print(arr[i] + ",");
        }
        System.out.println(" ");
        System.out.println("Enter the preferred sorting:");
        System.out.println("1.BubbleSort,2.SelectionSort");
        int a=sc.nextInt();
        switch(a) {
            case 1:
                BubbleSortORSelectionSort ob = new BubbleSortORSelectionSort();
                ob.bubbleSort(arr);
                System.out.println("Sorted array");
                ob.printArray(arr);
                break;
            case 2:
                BubbleSortORSelectionSort obj = new BubbleSortORSelectionSort();

                obj.sort(arr);
                System.out.println("Sorted array");
                obj.printArray(arr);
                break;
        }
    }
}

```

OUTPUT



```

<terminated> BubbleSortORSelectionSort [Java Application] C:\Users\HP\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.1.v2021
Enter the number of integers we want to enter
5
Enter the number of elements
20
4
58
100
69
The array elements are
20,4,58,100,69,
Enter the preferred sorting:
1.BubbleSort,2.SelectionSort
1
Sorted array
4 20 58 69 100

```

```
Console X
<terminated> BubbleSortORSelectionSort [Java Application] C:\Users\HP\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.1.v2021
Enter the number of integers we want to enter
5
Enter the number of elements
56
4
20
100
85
The array elements are
56,4,20,100,85,
Enter the preferred sorting:
1.BubbleSort,2.SelectionSort
2
Sorted array
4 20 56 85 100
```

QUESTION 2

Write a program to implement insertion sort.

package Assignment;

public class InsertionSort {

public static void main(String[] args) {

 System.out.println("Before Insertion Sort");

int a[] = { 25, 55, 2, 90, 45 };

int temp, j;

for (**int** i = 1; i < a.length; i++) {

 temp = a[i];

 j = i;

while (j > 0 && a[j - 1] > temp) {

 a[j] = a[j - 1];

 j = j - 1;

 }

 a[j] = temp;

for (**int** k = 0; k < a.length; ++k) {

 System.out.print(a[k] + " ");

 }

 System.out.println();

 }

for (**int** i = 0; i < a.length; i++) {

 System.out.print(a[i] + " ");

 }

 System.out.println();

 System.out.println("After Insertion Sort");

for (**int** i : a) {

 System.out.print(i + " ");

 }

 }

}

OUTPUT

```
<terminated> InsertionSort [Java Application] C:\Users\HP\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.1.v20211116-1657\jre\l
Before Insertion Sort
25 55 2 90 45
2 25 55 90 45
2 25 55 90 45
2 25 45 55 90
2 25 45 55 90
After Insertion Sort
2 25 45 55 90
```

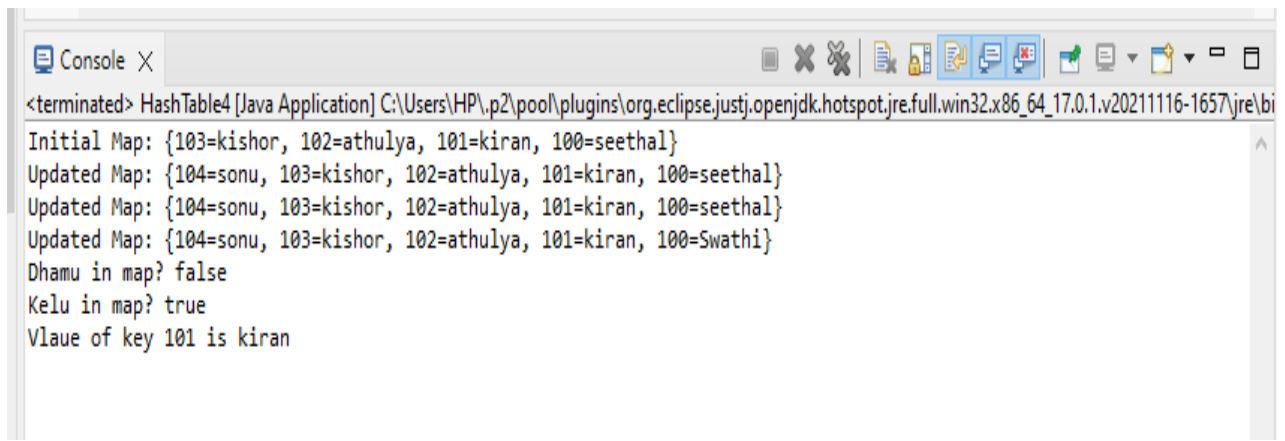
QUESTION 3

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

```
package hashtable;
import java.util.Hashtable;

public class HashTable4 {
    public static void main(String[] args) {
        Hashtable<Integer, String> m = new Hashtable<Integer, String>();
        m.put(100, "seethal");
        m.put(102, "athulya");
        m.put(101, "kiran");
        m.put(103, "kishor");
        System.out.println("Initial Map: " + m);
        // Inserts, as the specified pair is unique
        m.putIfAbsent(104, "sonu");
        System.out.println("Updated Map: " + m);
        // Returns the current value, as the specified pair already exist
        m.putIfAbsent(101, "Vijay");
        System.out.println("Updated Map: " + m);
        // Replace the value at key 100
        m.replace(100, "Swathi");
        System.out.println("Updated Map: " + m);
        // Checking values in map
        System.out.println("Dhamu in map? " + m.contains("sivadas"));
        System.out.println("Kelu in map? " + m.contains("Swathi"));
        // Checking key in map and getting the value
        if (m.containsKey(101) == true) {
            System.out.println("Vlaue of key 101 is " + m.get(101));
        }
    }
}
```

OUTPUT



```
<terminated> HashTable4 [Java Application] C:\Users\HP\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.1.v20211116-1657\jre\bin\java.exe
Initial Map: {103=kishor, 102=athulya, 101=kiran, 100=seethal}
Updated Map: {104=sonu, 103=kishor, 102=athulya, 101=kiran, 100=seethal}
Updated Map: {104=sonu, 103=kishor, 102=athulya, 101=kiran, 100=seethal}
Updated Map: {104=sonu, 103=kishor, 102=athulya, 101=kiran, 100=Swathi}
Dhamu in map? false
Kelu in map? true
Vlaue of key 101 is kiran
```

QUESTION 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.

```
package hashtable;
import java.util.Hashtable;
import java.util.Map;
class Book {
    int id;
    String name, author, publisher;
    int quantity;

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

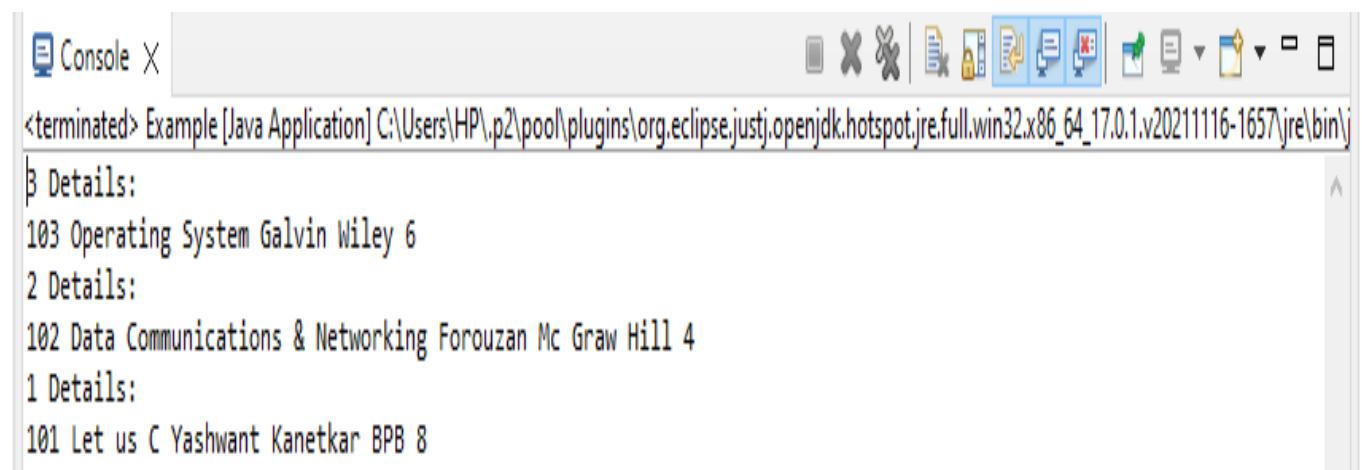
    public static void main(String[] args) {
        // Creating map of Books
        Hashtable<Integer, Book> map = new Hashtable<Integer, Book>();
        // Creating Books
```

```

    Book b1 = new Book(101, "Let us C", "Yashwant Kanetkar", "BPB", 8);
    Book b2 = new Book(102, "Data Communications & Networking",
"Forouzan", "Mc Graw Hill", 4);
    Book b3 = new Book(103, "Operating System", "Galvin", "Wiley", 6);
    // 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);
    }
}

```

OUTPUT



```

<terminated> Example [Java Application] C:\Users\HP\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.1.v20211116-1657\jre\bin\j
3 Details:
103 Operating System Galvin Wiley 6
2 Details:
102 Data Communications & Networking Forouzan Mc Graw Hill 4
1 Details:
101 Let us C Yashwant Kanetkar BPB 8

```