CSE_CS104_Practice2_2019

Test Summary

- No. of Sections: 2
- No. of Questions: 43
- Total Duration: 11000 min

Section 1 - MCQs

Section Summary

- No. of Questions: 31
- Duration: 1000 min

Additional Instructions:

None

Q1. Which of the following condition is for array of objects?

- All of the objects must be of different class
- All of the objects must be of same class
- At least some of the objects must be of same program classes
- All of the objects must have different data

Q2. Which component is responsible for converting bytecode into machine specific code?

- JVM
- JDK
- JRE
- JIT

Q3.

```
Which option will be correct for following code?

class Student {

  static int j = 1;

public static void main(String[] args)

  {

    for (int j = 1; j < 10; j++) {

      j = j + 2;

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

    }

  }
}
```

3 6 9....27

None of these

Run Time Error

3 6 9

Q4. We have z = 0, which of the following are legal lines of Java code for changing the value of z to 1?

1. z++;
2. z = z + 1;
3. z += 1;
4. z =+ 1;

a) 1 & 4

b) 1 ,2&3

c) All

d) 3 & 2

Q5. Choose the correct output

```

Class File{

public static void main(String args[])

{

int z;

System.out.println(z);

}

}
```

a)0

b)garbage value

c) Run time error

d) Compile time error

Q6. **Which of the following is not a feature of JAVA**

1)Garbage Collection

2)Portable

3)Secure

4)Platform Dependent

Q7. **Which of the following is not true.**

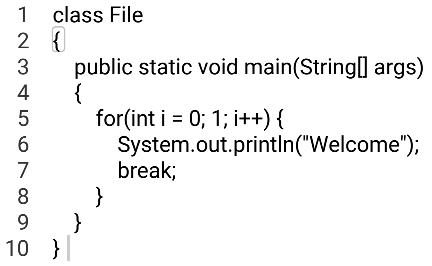
a) Enhanced for loops are not appropriate when you want to modify the array.

b) For each loop (also called as Enhanced for loop) keeps track of index of an array.

c) Enhanced for loop only iterates forward over the array in single steps

d) Enhanced for loop starts with a keyword “for” like a normal for loop

Q8. Choose the correct option as output for the following code



```
class File
{
   public static void main(String[] args)
   {
      for(int i = 0; 1; i++) {
         System.out.println("Welcome");
         break;
      }
   }
}
```

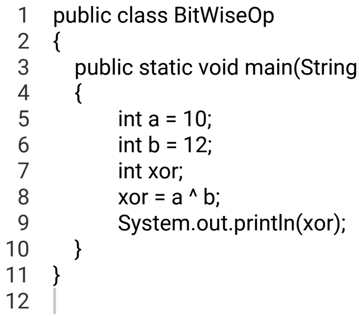
a)Compile time error

b)Welcome

c)infinite loop printing Welcome

d)None of these

Q9. What is the output for the below Java code ?



```
public class BitWiseOp
{
   public static void main(String[] args)
   {
         int a = 10;
         int b = 12;
         int xor;
         xor = a ^ b;
         System.out.println(xor);
   }
}
```

a) 8

b) 14

c) 6

d) -11

Q10. What is the output for the below Java code ?

```
public class OperatorsExample
{
  public static void main(String[] args)
{
      boolean b;
      int j, k;

      j = 0; k = 0;
      b = ( j++ == k ) && ( j == ++k );
      System.out.println("b, j, k " + b + ", " + j + ", " + k);
      j = 0; k = 0;
      b = ( j++ != k ) && ( j == ++k );
      System.out.println("b, j, k " + b + ", " + j + ", " + k);
  }
}

```

b, j, k true, 0, 0
b, j, k false, 1, 1

b, j, k true, 1, 1
b, j, k false, 1, 0

b, j, k false, 0, 0
b, j, k true, 1, 1

b, j, k false, 0, 0
b, j, k false, 1, 1

Q11. What is the output for the below Java code?

```
 1  public class Test
 2  {
 3    public static void main(String[] args)
 4  {
 5      int i = 010;
 6      int j = 07;
 7      System.out.println(i);
 8      System.out.println(j);
 9    }
10  }
11
```

a) 8 7

b) 10 7

c) Compilation fails with an error at line 3

d) Compilation fails with an error at line 5

Q12. Which of the following is not a primitive datatype in java?

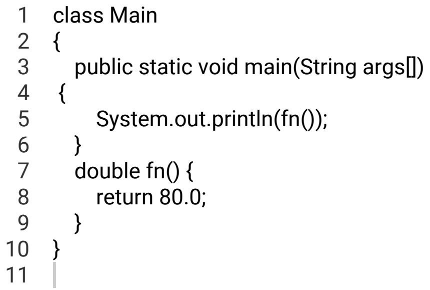
a) short

b) int

c) String

d) boolean

Q13. What will be the output of the following code?



```
class Main
{
  public static void main(String args[])
 {
    System.out.println(fn());
  }
  double fn() {
    return 80.0;
  }
}
```

a) 80.0

b) fn()

c) 80

d) compiler error

Q14. What is the stored in the object obj in following lines of code?
box obj;

Memory address of allocated memory of object

NULL

Garbage

Any arbitrary pointer

Q15. Which of these is an incorrect array declaration?

a) int arr[] = new int[5].

b) int [] arr = new int[5].

c) int arr[] = new int[5].

d) int arr[] = int [5] new.

Q16.

```
Predict the output of following code:

class Demo {
  public static void main(String args[]) {
     System.out.println(fun());
  }
 static int fun() {
     static int x= 0;
     return ++x;
  }
}
```

1

0

Compilation Error

Run time Error

Q17. Which of these cannot be declared static?

class

object

method

variable

Q18.

```
Predict the output of the following java code

class Test {
 static int x=9;
 int fun1() {
   return x++;
 }
 public static void main(String args[]) {
   Test m=new Test();
   x=m.fun1();
   System.out.println(x);
 }
}
```

9

10

0

error

Q19. Predict the output of the following code

```
class A
{ int a,b;
  A()
  {
   a=1;
  }
  A(int x)
  {
   this();
   b=x;
   B b=new B(this);
  }
}
class B
{
  A a1;
  B(A a1 )
  {
    this.a1=a1;
    display(this);
  }
   void display(B x)
  {
    int y=this.a1.a;
    y-=this.a1.b;
    System.out.println(y);
  }

}
class Main
  {
    public static void main(String args[])
      {
    A a=new A(10);
      }
  }
```

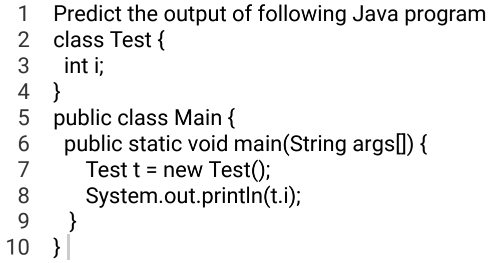
9

-9

10

1

Q20.



Predict the output of following Java program

```
class Test {
  int i;
}
public class Main {
  public static void main(String args[]) {
      Test t = new Test();
      System.out.println(t.i);
   }
}
```

garbage value

0

compiler error

runtime error

Q21. What would be behaviour if the constructor has a return type?

Compilation error

Runtime error

Compilation and runs successfully

None of the above

Q22.

```
What will be the output of the following program?

class ArrayOutput
{
   public static void main(String s[])
   {
      int a[] = {12, 15, 16, 17, 19};

      for(int i = 5 - 1; i > 0; i--)
      {
         System.out.println(a[i]);
      }
   }
}
```

```
17
16
15
12
```

```
19
17
16
15
12
```

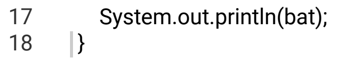
Compilation Error

```
19
17
16
15
```

Q23.

```
What will be the output of the following program?

public class Mcq1 {
   static boolean ball;
   public static void main(String[] args) {
      short bat = 42;
      if (bat < 50 & !ball)
         bat++;
      if (bat > 50)
         ;
      else if (bat > 40) {
         bat += 7;
         bat++;
      }
      else
         --bat;
```



```
        System.out.println(bat);
    }
```

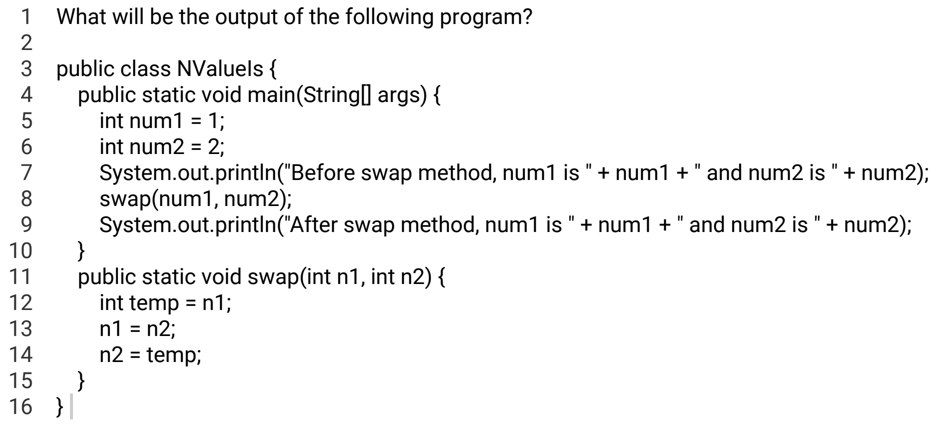
41

42

50

51

Q24.



What will be the output of the following program?

```
public class NValueIs {
    public static void main(String[] args) {
        int num1 = 1;
        int num2 = 2;
        System.out.println("Before swap method, num1 is " + num1 + " and num2 is " + num2);
        swap(num1, num2);
        System.out.println("After swap method, num1 is " + num1 + " and num2 is " + num2);
    }
    public static void swap(int n1, int n2) {
        int temp = n1;
        n1 = n2;
        n2 = temp;
    }
}
```

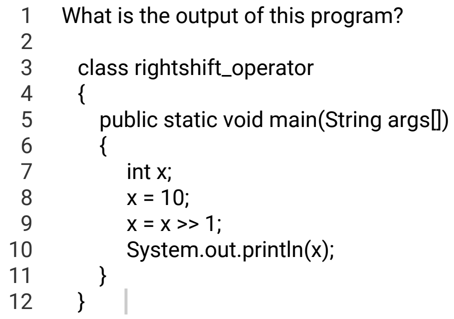
Before swap method, num1 is 1 and num2 is 2
After swap method, num1 is 2 and num2 is 1

Before swap method, num1 is 1 and num2 is 2
After swap method, num1 is 1 and num2 is 2

Compilation Error

Runtime Error

Q25.



What is the output of this program?

```
    class rightshift_operator
    {
        public static void main(String args[])
        {
            int x;
            x = 10;
            x = x >> 1;
            System.out.println(x);
        }
    }
```

10

5

2

20

Q26.

What is the output of this program?

```
 class evaluate
 {
     public static void main(String args[])
         {
         int arr[] = new int[] {0 , 1, 2, 3, 4, 5, 6, 7, 8, 9};
         int n = 6;
             n = arr[arr[n] / 2];
         System.out.println(arr[n] / 2);
         }
 }
```

3

0

6

1

Q27.

What will be the output of the following program?

```
public class GuessCondition
{
    static int a = 40;
    public static void main(String args[])
    {
        System.out.print(a + " ");
        add();
        System.out.print(a);
    }
    static void add()
    {
        a = a + 40;
    }
}
```

40 40

0 0

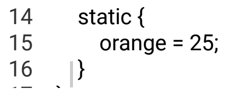
0 40

40 80

Q28.

What will be the output of the following program?

```
public class Mcq1 {
    static int orange, mango;
    static int banana;
    static {
        mango = orange + 3;
        banana = mango * 2;
    }
    public static void main(String[] args) {
        System.out.print(orange + " " + mango + " " + banana + " ");
        banana = mango + orange + banana;
        System.out.print(banana);
    }
```



```
    static {
       orange = 25;
    }
```

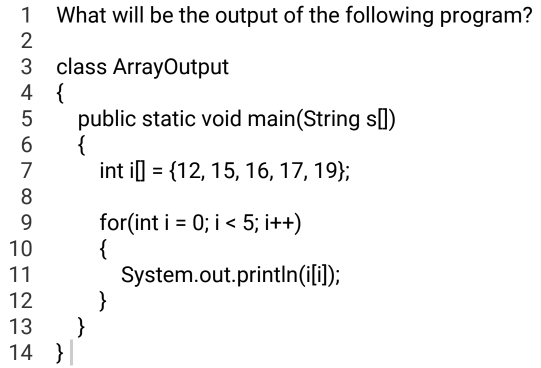
25 3 6 34

0 3 6 9

3 0 6 9

9 6 3 0

Q29.



```
What will be the output of the following program?

class ArrayOutput
{
   public static void main(String s[])
   {
      int i[] = {12, 15, 16, 17, 19};

      for(int i = 0; i < 5; i++)
      {
         System.out.println(i[i]);
      }
   }
}
```

```
12
15
16
17
19
```

```
12
15
16
17
```

```
15
16
17
19
```

Compilation Error

Q30.

Predict the output

```
class Test
{
 static int count=0;
  private Test()
{  count++;  System.out.println(count);
 }
 }
class Main
{
 public static void main(String[] args)
{
  b1:
 for(int i = 0; true; i++)
{
  Test c1=new Test();
  Test c2=new Test();
 Test c3=new Test();
  break b1;
```

```
        }
 }
}
```

```
1
2
3
```

```
0
1
2
```

Error

no output

Q31. Predict the output of the following code

```
class A
{
 A()
 {
  Main m=new Main(20);
  System.out.println(m.x);
 }
}
class Main
 {
  int x;
  Main()
  {
   this.x=10;
   System.out.println(this.x);
  }
  Main(int x)
  {
   this();
   this.x=x;
   System.out.println(this.x);
  }
  public static void main(String args[])
   {
  A a=new A();
   }
 }
```

```
10
10
20
```

```
20
20
```

```
10
20
20
```

```
10
20
```

Section 2 - Coding

Section Summary

- No. of Questions: 12
- Duration: 10000 min

Additional Instructions:

None

Q1. Ronit is very fond of horses. He enjoys watching them race. As expected, he has a stable full of horses. He, along with his friends, goes to his stable during the weekends to watch a few of these horses race. Ronit wants his friends to enjoy the race and so he wants the race to be close. This can happen only if the horses are comparable on their skill i.e. the difference in their skills is less.
There are **N** horses in the stable. The skill of the horse **i** is represented by an integer **S[i]**. The Ronit needs to pick 2 horses for the race such that the difference in their skills is *minimum*. This way, he would be able to host a very interesting race. Your task is to help him do this and report the minimum difference that is possible between 2 horses in the race.

Input Format

First line of input file contains a single integer T, the number of testcases.
Every testcase starts with a line containing the integer N.
The next line contains N space seperated integers where the i-th integer is S[i].

Output Format

For each test-case, output a single line containing the minimum difference that is possible.

Sample Input

```
1
5
4 9 1 32 13
```

Sample Output

```
3
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q2. MERGE TWO SORTED ARRAYS

Teacher has given you two sorted integer arrays A and B,and you have to perform a task of merging both array in a sorted manner.

***Note*: You have to complete the merge function and output merged array.**

Input Format

3 // size of First array
16
18
20
5 // size of Second array
11
15
16
18
20

Output Format

Tab separated sorted and merged array
11 15 16 16 18 18 20 20

Sample Input

```
4
2
5
6
```

Sample Output

```
2 5 6 6 8 8 9
```

Sample Input

```
2
5
8
1
```

Sample Output

```
5 8 9
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q3. Mr. Shibe is teaching Java to his students.In today's coding challenge he asked students to read an array as input and rotate the array towards left by k times.Solve Mr. Shibe's coding challenge by considering following illustration:

For example consider the array
1 2 3 4 5
let k is number of left rotations
for k=1
shift every element of the array towards left by one index and we are done.
so output will be 2 3 4 5 1
for k=2
shift every element towards left by 2 positions
so output will be 3 4 5 1 2

Input Format

First Line read the size of array say n
Next n line read n elements for the array.
Next line reads number of rotaions say k.

Output Format

output prints the array rotated k time towards left

Sample Input

```
5
1
2
3
```

Sample Output

```
3 4 5 1 2
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q4. ***Next Smaller Element***
Given an array, print the Next Smaller Element (NSE) for every element. The smaller element for an element x is the first smaller element on the right side of x in array. *Elements for which no smaller element exist (on right side), consider next smaller element as -1.*

For the input array {4, 8, 5, 2, 25}, the next smaller elements for each element are as follows.

```
Element      NSE
  4    -->  2
  8    -->  5
  5    -->  2
  2    -->  -1
  25    -->  -1
```

Input Format

Input size of the array, n.
Input n elements of the array.

Output Format

Output Next Smaller Element (NSE) for every element.

Sample Input

```
5
4
8
5
```

Sample Output

```
2
5
2
-1
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q5. ***Array Sum***
Harshit is a computer science student and his teacher gave him a simple question to solve within time but he is little busy in his life, so he asked you to solve this problem.
You have been given an array of positive integers A1,A2,...,An with length N and you have to print an array of same length(N) where the values in the new array are the sum of every number in the array, except the number at that index.

Input Format

First line contains N, the length of the array and second line contains N space separated positive integers.

Output Format

Output a single array of same length,in which the values in the new array are the sum of every number in the array, except the number at that index.

Sample Input

```
4
1 2 3 4
```

Sample Output

```
9 8 7 6
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q6. ***Next Greater Element***

Given an array, print the Next Greater Element (NGE) for every element. The Next greater Element for an element x is the first greater element on the right side of x in array. Elements for which no greater element exist, consider next greater element as -1.

For the input array {13, 7, 6, 12}, the next greater elements for each element are as follows.

```
 Element      NGE
  13    -->  -1
  7     -->   12
  6     -->   12
  12    -->   -1
```

Input Format

Input size of array, N
Input N, array elements

Output Format

Output the next greater elements for each element

Sample Input

```
4
13
7
6
```

Sample Output

```
-1
12
12
-1
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q7. ***Kaprekar Number***

A Kaprekar number is a number whose square when divided into two parts and such that sum of parts is equal to the original number and none of the parts has value 0.
Given a number, the task is to check if it is ***Kaprekar Number*** or not.

Example:
Input : n = 45
Output : Yes
Explanation : 45^2 = 2025 and 20 + 25 is 45

Input Format

In First line, enter no. of testcases, t.
Enter the number in next line for each test case.

Output Format

Output true if its is Kaprekar number else print false

Sample Input

```
2
45
13
```

Sample Output

```
true
false
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q8. An electricity board charges the following rates to demonstrate users to discourage large consumption of energy:
For the first 100 units -60P per unit
For next 200 units- 80P per unit
Beyond 300 units 90P per unit
If the total amount becomes more than Rs, 300.00 then additional 200 P are charged.
Write a program to read the number units consumed and print out the net charges.
(Here P denotes Paisa, and Rs denotes Rupee Indian currency) and 1 Rs=100 P)

Input Format

Read number of units consumed e.g.,
150

Output Format

Amount to be paid is: 10000P

Explanation:
For first 100 the amount is 100*60=6000
For next 50 the amount is 50*80=4000
So total amount becomes 6000+4000=10000

Sample Input

```
150
```

Sample Output

```
Amount to be paid is: 10000P
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q9. **Cut the Sticks**

You will be given an array of the lengths of a number of sticks. When a turn begins, you must count the number of sticks you have. Determine the length of a stick with the shortest length and discard any sticks of that length. Remove that length from each of the longer sticks and discard the offcuts.

Repeat until there are no sticks left. Return an array where the elements are the numbers of sticks you had at the beginning of each turn.

For example, consider an array representing the lengths of four sticks: [1, 1, 2, 3]. The shortest sticks are 1 unit long. Discard them. Remove 1 unit from the other two sticks and discard the scrap. Now you have two sticks lengths [1, 2]. Do the same and you'll have one stick of length [1]. Discard it and return an array with the number of sticks you had at the start of each turn: [4, 2, 1].

lengths	cut length	sticks
1 1 2 3	1	4
_ _ 1 2	1	2
_ _ _ 1	1	1
_ _ _ _	DONE	DONE

Input Format

The first line contains an integer *n*, the number elements in *lengths*.
The next *n* lines each represents *lengths[i]* where 0 ≤ i < n.

6 //number of inputs
5
4
4
2
2
8

Output Format

6
4
2
1

Constraints

- $1 \leq n \leq 10^3$
- $1 \leq lengths[i] \leq 10^3$, where $0 \leq i < n$

Sample Input

```
8
1
2
3
```

Sample Output

```
8
6
4
1
```

Sample Input

Sample Output

```
5
9
7
3
```

```
5
4
3
2
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q10. Jagged array is array of arrays or group of array within it such that member arrays can be of different sizes. It can be of any size or irregular multidimensional array. To make an array in jagged(each row will contain different no. of elements) form take given input from user.

Input

1. Integer **r** to enter the number of rows for an array
2. Integer **c** to enter the number of Elements in each row 1 to n .
3. Array arr[][] to enter the elements in an array.

Output:

1. Total no. of rows.
2. Display array in Matrix form.
3. Total number of elements in the array.
4. Calculate Average of the elements in the array.

Input Format

```
3
2
3
4
1
5
2
1
2
```

Output Format

```
Output
3 4
5
1 2
Total number of Elements in array-5
Average of the elements of the array-3
```

Sample Input

```
4
4
9
8
```

Sample Output

```
Output
9 8 7 6
3
5 6 7
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q11. **Problem: check Smartphone**

A company " Amfinix " launches new updates of fingerprint and 3 multi-window for smartphones in market. These updates are compatible with smartphones only so it need to develop a system which checks if the phone is a smartphone or not.

For this System to implement it sets a criteria depending on which it judges mobile phone has smartphone features.

following criteria is used:

if RAM used in phone is in range 512MB to 4 GB, range of display of phone is from 480p to 1440p and the processor in phone is one of the following:

dual core or quad core or octa core then the phone is considered as smartphone.

It creates a class "Amfinix" which takes basic information of mobile phone (like processor used, name, RAM used,display of phone,battery life provided by phone, phone is single sim or not, flashlight is provided or not) from class "mobile".Method check(), checks if phone is smartphone in interface "smartphone". Methods get_data() and display() present in both classes ,class Amfinix as well as in class mobile and take phone information from user and display phone information with the result of check() method. Amfinix class checks a phone to be a smartphone or not, and if the phone is smartphone then provide an update fingerprint and multi-window. After installation of these updates mobile provides fingerprint and 3 window option in multiwindow.

Input Format

```
first line gives the value of processor of mobile
second line provides the name of phone
thirsd line provides the RAM of mobile
forth line provides the value of display of mobile
fifth line provides the battery life
sixth line provides information whether phone is single sim
seventh line gives information if phone has flashlight
```

Output Format

```
first seven lines provides phone information
eight line provides information whether phone has fingerprint feature which is initially false
ninth line gives information about the number of windows phone support in multiwindow feature which is initially set to one
tenth line shows the result of check(), if phone is smartphone or not
if phone is smartphone update the phone and display information in next nine lines
```

Sample Input

```
dual core
samsung
512
480
```

Sample Output

```
dual core
samsung
512
480
```

Sample Input

```
dual core
samsung
512
42280
```

Sample Output

```
dual core
samsung
512
42280
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q12. Sorting is the way of arranging data elements in particular format. Sort an array of integers in ascending order by using **Selection sort** algorithm. Explanation of selection sort is given below to use in sorting integers.

Initial Array : 4, 5, 1, 25, 10

First iteration : Compare every element after first element with first element and if it is larger then swap. In first iteration, 4 is larger than 1. So, swap it. **1, 5, 4, 25, 10**

Second iteration : Compare every element after second element with second element and if it is larger then swap. In second iteration, 5 is larger than 4. So, swap it. **1, 4, 5, 25, 10**

Third iteration : Nothing will swap as 5 is smaller than every element after it.

1, 4, 5, 25, 10

Fourth iteration : Compare every element after fourth element with fourth element and if it is larger then swap. In fourth iteration, 25 is larger than 10. Swap it. **1, 4, 5, 10, 25**

Input Format

6
65
34
27
45
66
12

Output Format

Sorted Elements
12 27 34 45 65 66

Sample Input

```
4
5
1
25
```

Sample Output

```
Sorted Elements
1 5 10 25
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Answer Key & Solution

Section 1 - MCQs

Q1 All of the objects must be of same class

Solution

No Solution

Q2 JVM

Solution

No Solution

Q3 3 6 9

Solution

No Solution

Q4 c) All

Solution

No Solution

Q5 d) Compile time error

Solution

No Solution

Q6 4)Platform Dependent

Solution

No Solution

Q7 b) For each loop (also called as Enhanced for loop) keeps track of index of an array.

Solution

No Solution

Q8 a)Compile time error

Solution

No Solution

Q9 c) 6

Solution

No Solution

Q10 b, j, k true, 1, 1

b, j, k false, 1, 0

Solution

No Solution

Q11 a) 8 7

Solution

No Solution

Q12 c) String

Solution

No Solution

Q13 d) compiler error

Solution

No Solution

Q14 NULL

Solution

No Solution

Q15 d) int arr[] = int [5] new.

Solution

No Solution

Q16 Compilation Error

Solution

No Solution

Q17 object

Solution

No Solution

Q18 9

Solution

No Solution

Q19 -9

Solution

No Solution

Q20 0

Solution

No Solution

Q21 Compilation error

Solution

No Solution

Q22 19

17

16

15

Solution

No Solution

Q23 51

Solution

No Solution

Q24 Before swap method, num1 is 1 and num2 is 2

After swap method, num1 is 1 and num2 is 2

Solution

No Solution

Q25 5

Solution

No Solution

Q26 1

Solution

No Solution

Q27 40 80

Solution

No Solution

Q28 25 3 6 34

Solution

No Solution

Q29 Compilation Error

Solution

No Solution

Q30 Error

Solution

No Solution

Q31 10

20

20

Solution

No Solution

Section 2 - Coding

Q1 **Test Case**

Input

```
2
3
2 1 4
6
```

Output

```
1
1
```

Weightage - 20

Input

```
3
2
2 4
4
```

Output

```
2
1
1
```

Weightage - 20

Input

```
3
3
3 5 7
2
```

Output

```
2
3
0
```

Weightage - 20

Input

```
2
3
21 56 27
2
```

Output

```
6
0
```

Weightage - 20

Input

```
1
6
21 89 88 24 10 15
```

Output

```
1
```

Weightage - 20

Sample Input

```
1
5
4 9 1 32 13
```

Sample Output

```
3
```

Solution

Solution

Header

```
import java.util.*;
import java.lang.*;
import java.io.*;

public class file
{



public static void main (String[] args) throws java.lang.Exception
    {
        int t;
        Scanner sc =  new Scanner(System.in);
        t  = sc.nextInt();
        while(t>0)
        {
            int n,i;
            n = sc.nextInt();
            int a[] = new int[n];
            for(i =0;i<n;i++)
            {
                a[i] = sc.nextInt();
            }
            Arrays.sort(a);
            int m = a[1]-a[0];
            for(i=2;i<=n-1;i++)
            {
             if(m>a[i]-a[i-1])
             {
             m = a[i]-a[i-1];
            }
            }
            System.out.println(m);
            t--;
        }
    }
```

Footer

```
}
```

Q2 Test Case

Input

```
7
11
15
16
```

Output

```
11 12 15 16 16 18 18 19 20 22 22 24
```

Weightage - 20

Input

Output

```
8
22
25
28
```

```
22 25 28 33 34 36 37 38 40 44 44 47
```

Weightage - 20

Input

Output

```
10
22
35
37
```

```
12 16 18 22 22 35 37 44 47 49 50 57 59 60
```

Weightage - 20

Input

Output

```
6
33
37
40
```

```
33 36 37 38 40 40 46 49 50 50 60 69
```

Weightage - 20

Input

Output

```
1
4
5
2
```

```
2 4 6 8 9 10
```

Weightage - 20

Sample Input

Sample Output

```
4
2
5
6
```

```
2 5 6 6 8 8 9
```

Sample Input

Sample Output

```
2
5
8
1
```

```
5 8 9
```

Solution

Header

```
import java.util.*;
public class file
{
    static void merge(int []A,int []B)
    {



int [] R=new int[A.length+B.length];
        int i=0;
        int k=0, j=0;

        while(k<A.length && j<B.length)
        {
            if(A[k]<=B[j])
            {

                R[i] = A[k];
                k++;
                i++;

            }
            else
            {
                R[i] = B[j];
                j++;
                i++;
            }
        }

         while(k<A.length)
        {
                R[i] = A[k];
                k++;
                i++;
        }

         while(j<B.length)
        {
                R[i] = B[j];
                j++;
                i++;
        }
int n1=A.length;
int n2=B.length;
//System.out.println(n1+" "+n2);
for(i=0;i<n1+n2;i++)
    {
        System.out.print(R[i]+" ");
    }
```

Footer

```
}
public static void main(String args [])
{

Scanner sc = new Scanner(System.in);
int n1=sc.nextInt();
int [] A=new int[n1];

 for(int i=0;i<n1;i++)
    {
        A[i]=sc.nextInt();
    }
int n2=sc.nextInt();
int [] B=new int[n2];
for(int i=0;i<n2;i++)
    {
        B[i]=sc.nextInt();
    }
```

```
merge(A,B);
}
}
```

Q3 Test Case

Input

```
5
1
2
3
```

Output

```
1 2 3 4 5
```

Weightage - 20

Input

```
6
1
2
4
```

Output

```
5 6 7 1 2 4
```

Weightage - 20

Input

```
10
11
12
13
```

Output

```
20 11 12 13 14 15 16 17 18 19
```

Weightage - 20

Input

```
1
1
2
```

Output

```
1
```

Weightage - 20

Input

```
2
1000
200
0
```

Output

```
1000 200
```

Weightage - 20

Sample Input

```
5
1
2
3
```

Sample Output

```
3 4 5 1 2
```

Solution

Header

```
import java.util.*;



class RotateArray {

    void leftRotate(int arr[], int d, int n)
    {
            for (int i = 0; i < d; i++)
            leftRotatebyOne(arr, n);
    }

    void leftRotatebyOne(int arr[], int n)
    {
        int i, temp;
        temp = arr[0];
        for (i = 0; i < n - 1; i++)
            arr[i] = arr[i + 1];
        arr[i] = temp;
    }


    void printArray(int arr[], int n)
    {
            for (int i = 0; i < n; i++)
            System.out.print(arr[i] + " ");
    }
}
```

Footer

```
public class file{

    public static void main(String[] args)
    {
        int rotations;
        Scanner sc=new Scanner(System.in);
        int arraysize=sc.nextInt();

        int arr[]=new int[arraysize];
        for(int i=0;i<arraysize;i++)
        arr[i]=sc.nextInt();

        rotations=sc.nextInt();
        RotateArray rotate = new RotateArray();

        rotate.leftRotate(arr, rotations, arraysize);
        rotate.printArray(arr,arraysize);
    }
}
```

Q4 Test Case

Input

```
5
2
3
2
```

Output

```
1
2
1
-1
```

Weightage - 20

Input

```
4
25
22
21
```

Output

```
22
21
20
-1
```

Weightage - 20

Input

```
6
12
-12
13
```

Output

```
-12
-13
-13
-14
```

Weightage - 20

Input

```
5
5
4
3
```

Output

```
4
3
2
1
```

Weightage - 20

Input

```
3
213
-213
213
```

Output

```
-213
-1
-1
```

Weightage - 20

Sample Input

```
5
4
8
5
```

Sample Output

```
2
5
2
-1
```

Solution

Header

```
import java.util.*;
public class file {
    static void printNSE(int arr[], int n)
    {
        int next, i, j;
        for (i = 0; i < n; i++) {
            next = -1;
            for (j = i + 1; j < n; j++) {
                if (arr[i] > arr[j]) {
                    next = arr[j];
                    break;
                }
            }
            System.out.println(next);
        }
```

Footer

```
    }

    public static void main(String args[])
    {
        Scanner sc=new Scanner(System.in);
        int n=sc.nextInt();
        int arr[]=new int[n];
        for(int i=0;i<n;i++)
            arr[i]=sc.nextInt();
        printNSE(arr, n);
    }
}
```

Q5 Test Case

Input

```
5
0 0 0 0 0
```

Output

```
0 0 0 0 0
```

Weightage - 20

Input

```
3
-1 -1 -1
```

Output

```
-2 -2 -2
```

Weightage - 20

Input

```
5
1 1 3 4 0
```

Output

```
8 8 6 5 9
```

Weightage - 20

Input

```
4
0 -1 1 -2
```

Output

```
-2 -1 -3 0
```

Weightage - 20

Input

```
3
100 1000 10000
```

Output

```
11000 10100 1100
```

Weightage - 20

Sample Input

```
4
1 2 3 4
```

Sample Output

```
9 8 7 6
```

Solution

Header

```
import java.util.*;
public class file
{
    static void arraySum(int[] ar)
    {



        int sum=0;
         for(int i=0;i<ar.length;i++){
            sum+=ar[i];
         }
         for(int i=0;i<ar.length;i++)
         {
           System.out.print((sum-ar[i])+" ");
         }
```

Footer

```
    }
    public static void main (String[] args)
    {
        Scanner ag=new Scanner(System.in);
         int n=ag.nextInt();
         int[] ar=new int[n];
         for(int i=0;i<n;i++){
            ar[i]=ag.nextInt();
         }
         arraySum(ar);
    }
}
```

Q6 **Test Case**

Input

```
5
1
2
3
```

Output

```
2
3
4
5
```

Weightage - 20

Input

```
5
5
4
3
```

Output

```
-1
-1
-1
-1
```

Weightage - 20

Input

```
6
6
7
7
```

Output

```
7
8
8
9
```

Weightage - 20

Input

```
4
123
123
123
```

Output

```
-1
-1
-1
-1
```

Weightage - 20

Input

```
2
-6789
-4567
```

Output

```
-4567
-1
```

Weightage - 20

Sample Input

```
4
13
7
6
```

Sample Output

```
-1
12
12
-1
```

Solution

Header

```
import java.util.*;
public class file
{
   static void printNGE(int arr[], int n)
   {



    int next, i, j;
       for (i=0; i<n; i++)
       {
           next = -1;
           for (j = i+1; j<n; j++)
           {
               if (arr[i] < arr[j])
               {
                   next = arr[j];
                   break;
               }
           }
           System.out.println(next);
       }
   }
```

Footer

```
    public static void main (String[] args)
    {
        Scanner ag=new Scanner(System.in);
         int n=ag.nextInt();
         int[] ar=new int[n];
         for(int i=0;i<n;i++){
            ar[i]=ag.nextInt();
         }
         printNGE(ar, n);
    }
}
```

Q7 **Test Case**

Input

```
2
13
45
```

Output

```
false
true
```

Weightage - 20

Input

```
3
45
13
10
```

Output

```
true
false
false
```

Weightage - 30

Input

Output

```
3
45
13
297
```

```
true
false
true
```

Weightage - 30

Input

Output

```
4
1
13
10
```

```
true
false
false
true
```

Weightage - 20

Sample Input

Sample Output

```
2
45
13
```

```
true
false
```

Solution

Header

```
import java.util.*;
public class file
{
    static boolean iskaprekar(int n)
    {



    if (n == 1)
           return true;
        int sq_n = n * n;
        int count_digits = 0;
        while (sq_n != 0)
        {
            count_digits++;
            sq_n /= 10;
        }
        sq_n = n*n;
        for (int r_digits=1; r_digits<count_digits; r_digits++)
        {
             int eq_parts = (int) Math.pow(10, r_digits);
             if (eq_parts == n)
                continue;
             int sum = sq_n/eq_parts + sq_n % eq_parts;
             if (sum == n)
               return true;
        }
        return false;
    }
```

Footer

```
    public static void main (String[] args)
    {
        Scanner sc=new Scanner(System.in);
        int t=sc.nextInt();
        for (int i=t; i>=1; i--) {
            int num=sc.nextInt();
             System.out.println(iskaprekar(num));
        }
    }
}
```

Q8 **Test Case**

Input	Output
600	Amount to be paid is: 49200P

Weightage - 20

Input	Output
555	Amount to be paid is: 45150P

Weightage - 20

Input	Output
490	Amount to be paid is: 39300P

Weightage - 20

Input	Output
200	Amount to be paid is: 14000P

Weightage - 20

Input	Output
330	Amount to be paid is: 24700P

Weightage - 20

Sample Input	Sample Output
150	Amount to be paid is: 10000P

Solution

Header

```
import java.util.*;
class Bill
{



   int units;
   Bill(int a)
   {
      units=a;
   }
 int compute_bill()
{
      int bill=0;
        if(units<=100)
       {                 bill=units*60;

       }
        else if(units>100 && units<=200)
     {              bill=60*100;
                     units=units-100;
                      bill=bill+units*80;

      }
       else if(units>200)
     {
    bill=60*100+80*200;
                     units=units-300;
                      bill=bill+units*90;

       }
 if(bill>30000)
{  bill=bill+200;
}
                      return bill;
   }

}
```

Footer

```
public class file
{
    public static void main(String args[])
      {
              Scanner sc=new Scanner(System.in);
              int n;
              n=sc.nextInt();
              Bill b=new Bill(n);
              int amount=   b.compute_bill();
              System.out.println("Amount to be paid is: "+amount+"P");
```

```
        }
    }
```

Q9 Test Case

Input

```
6
5
4
4
```

Output

```
6
4
2
1
```

Weightage - 20

Input

```
8
1
4
1
```

Output

```
8
5
4
2
```

Weightage - 20

Input

```
10
11
16
19
```

Output

```
10
7
4
2
```

Weightage - 20

Input

```
10
25
30
22
```

Output

```
10
8
6
4
```

Weightage - 20

Input

```
15
33
54
43
```

Output

```
15
13
11
10
```

Weightage - 20

Sample Input

```
8
1
2
3
```

Sample Output

```
8
6
4
1
```

Sample Input

```
5
0
```

Sample Output

```
5
4
```

```
9
7
3
```

```
4
3
2
```

Solution

Header

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

```
    public static void main(String[] args) {

        Scanner cin = new Scanner(System.in);
        int n = cin.nextInt();
        int arr[] = new int[n];

        for(int i = 0; i < n; i++)
            arr[i] = cin.nextInt();

        Arrays.sort(arr);
        int count = 0;
        int index;
        for(int i = 0; i < n; )
        {
            index = i;
            while(i < n - 1 && arr[i] == arr[i + 1])
            {
                i++;
            }
            System.out.println(n - index);
            i++;
        }
    }
```

Footer

```
}
```

Q10 Test Case

Input

```
3
4
9
8
```

Output

```
Output
9 8 7 6
1 2
5 6 7
```

Weightage - 25

Input

```
3
3
```

Output

```
Output
22 43 56
```

```
22
```

```
6 45 55
```

Weightage - 25

Input

```
4
3
8
9
```

Output

```
Output
8 9 7
10 15
5 8 6 9
```

Weightage - 25

Input

```
5
1
10
3
```

Output

```
Output
10
9 6 3
4 7
```

Weightage - 25

Sample Input

```
4
4
9
8
```

Sample Output

```
Output
9 8 7 6
3
5 6 7
```

Solution

Header

```
import java.util.*;
public class file {
    public static void main(String[] args) {
          Scanner ob= new Scanner(System.in);




    int c=0,i, j, count=0, sum=0;

          int n= ob.nextInt();
          int arr[][]=new int[n][];

           for( i = 0; i<n;i++)
              {
                //System.out.println((i+1));
                c= ob.nextInt();
                count+=c;
                arr[i]=new int[c];
                for( j = 0;j<c;j++)
                 {
                    ob.nextLine();
                    //System.out.println(j+1);
                    arr[i][j] = ob.nextInt();

                    sum = sum + arr[i][j];
                 }
               }     System.out.println("Output");
               for(i = 0;i<n;i++){
```

```
          for( j = 0;j<arr[i].length;j++){
          System.out.print(arr[i][j]+" ");
          }
          System.out.println();

              }
      System.out.println("Total number of Elements in array-"+ count);
      System.out.println("Average of the elements of the array-"+ sum/count);
```

Footer

```
      }
  }
```

Q11 **Test Case**

Input

```
dual core
samsung
512
480
```

Output

```
dual core
samsung
512
480
```

Weightage - 20

Input

```
octa core
samsung
512
480
```

Output

```
octa core
samsung
512
480
```

Weightage - 20

Input

```
quad core
samsung
512
480
```

Output

```
quad core
samsung
512
480
```

Weightage - 20

Input

```
dual core
samsung
512
42280
```

Output

```
dual core
samsung
512
42280
```

Weightage - 20

Input

```
dual core
samsung
512
42280
```

Output

```
dual core
samsung
512
42280
```

Weightage - 20

Sample Input

```
dual core
samsung
512
480
```

Sample Output

```
dual core
samsung
512
480
```

Sample Input

```
dual core
samsung
512
42280
```

Sample Output

```
dual core
samsung
512
42280
```

Solution

Header

```
import java.util.*;
interface smartphone
{



 int display1_max=1440;
    int display1_min=480;
    int ram1_max=4000;
    int ram1_min=512;
    String p1=new String("dual core");
    String p2=new String("quad core");
    String p3=new String("octa core");
    void check();
}
class mobile
{
    String processor=new String();
    String name=new String();
    int RAM,display,battery;
    boolean isSingleSim,flashlight;
    void get_data()
    {

    }
    void display()
    {
        System.out.println(processor);
        System.out.println(name);
        System.out.println(RAM);
        System.out.println(display);
        System.out.println(battery);
        System.out.println(isSingleSim);
        System.out.println(flashlight);
    }
}
class Amfinix extends mobile implements smartphone
{
  boolean fingerprint=false;
    int multiwindow=1;
    void get_data()
    {
```

```
    {
        super.get_data();
          Scanner o=new Scanner(System.in);
        processor=o.nextLine();
        name=o.nextLine();
        RAM=o.nextInt();
        display=o.nextInt();
       battery=o.nextInt();
        isSingleSim=o.nextBoolean();
       flashlight=o.nextBoolean();
             //  fingerprint=o.nextBoolean();
     // multiwindow=o.nextInt();
     // camera=o.nextInt();
    }
    void display()
    {
        super.display();
              System.out.println(fingerprint);
        System.out.println(multiwindow);
      //  System.out.println(camera);
    }
    public void check()
    {
     if(processor.equals(p1) || processor.equals(p2)|| processor.equals(p3))
        {
            if(display<=1440 && display>=480 )
            if(RAM<=4000 && RAM>=512){
            System.out.println("smart phone"); fingerprint=true;
       multiwindow=3;
            display();
            }
        }
         else System.out.println("not smartphone");
    }
}
```

Footer

```
public class file
{
    public static void main(String[] args)
    {
        Amfinix a=new Amfinix();
        mobile m=a;
        smartphone h=a;
        m.get_data();
        a.display();
          h.check();
    }
}
```

Q12 Test Case

Input

```
7
23
45
```

Output

```
Sorted Elements
10 12 15 16 23 32 45
```

Weightage - 50

Input

```
6
45
34
65
```

Output

```
Sorted Elements
14 18 25 34 45 65
```

Weightage - 50

Sample Input

```
4
5
1
25
```

Sample Output

```
Sorted Elements
1 5 10 25
```

Solution

Header

```
import java.util.*;
public class file {
     public static void main(String[] args)
     {
```

```
       Scanner ob= new Scanner(System.in);
       int r = ob.nextInt();
       int arr[] = new int[r];
        int t=0;
        for(int i = 0;i<arr.length;i++){
            arr[i] = ob.nextInt();
        }
        System.out.println("Sorted Elements");
             for(int i=0; i<arr.length; i++)
             {
               for(int j = i+1;j<arr.length;j++){
            if(arr[i]>arr[j]){
              t = arr[i];
              arr[i] = arr[j];
              arr[j] = t;
            }
          }
        }
        for(int i = 0;i<arr.length;i++){
          System.out.print(arr[i]+" ");
        }
```

Footer

```
    }
}
```