1) What is the range of byte data type in Java?

**a) -128 to 127**

b) -32768 to 32767

c) -2147483648 to 2147483647

d) None of the mentioned

2) Which of these is a super class of all errors and exceptions in the Java language?

a) RunTimeExceptions

**b) Throwable**

c) Catchable

d) None of the above

3) Which of the following is correct way of implementing an interface salary by class manager?

a) class manager extends salary {}

**b) class manager implements salary {}**

c) class manager imports salary {}

d) None of the mentioned.

4) What is the output of this program?

class output

{

public static void main(String args[])

{

String s1 = "Hello";

String s2 = s1.replace('l','w');

System.out.println(s2);

}

}

a) Hello

b) Hewlo

c) Helwo

**d) Hewwo**

1. What will this code print?

**int** arr[] = **new** **int** [5];

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

System.***out***.print(arr[i]);

1. 0
2. value stored in arr[0].
3. **00000**
4. Class name@ hashcode in hexadecimal form
5. When Overloading does not occur?
6. More than one method with same name but different method signature and different number or type of parameters
7. More than one method with same name, same signature but different number of signature
8. More than one method with same name, same signature, same number of parameters but different type
9. **More than one method with same name, same number of parameters and type but different signature**
10. Which of the below is invalid identifier with main method?
11. Public
12. Static
13. **Private**
14. Final
15. How can a protected modifier be accessed?
16. accessible only within the class
17. **accessible within package and outside the package but through inheritance only**
18. accessible by all
19. accessible only within package
20. How many copies of static and class variables are created when 10 objects are created of a class?
21. **1,10**
22. 10,10
23. 10,1
24. 1,1
25. Predict the output of following Java program

class Main {

public static void main(String args[]) {

try {

throw 10;

}

catch(int e) {

System.out.println("Got the Exception " + e);

}

}

}

1. Got the Exception 10
2. Got the Exception 0
3. **Compiler Error**
4. None of the above
5. What is the output of this program?

class A

{

final public int calculate(int a, int b) { return 1; }

}

class B extends A

{

public int calculate(int a, int b) { return 2; }

}

public class output

{

public static void main(String args[])

{

B object = new B();

System.out.print("b is " + b.calculate(0, 1));

}

}

1. b is : 2
2. b is : 1
3. **Compilation Error.**
4. An exception is thrown at runtime.
5. What is the output of this program, if we run as “java main\_arguments 1 2 3”?

public class main\_arguments

{

public static void main(String [] args)

{

String [][] argument = new String[2][2];

int x;

argument[0] = args;

x = argument[0].length;

for (int y = 0; y < x; y++)

System.out.print(" " + argument[0][y]);

}

}

1. 1 1
2. 1 0
3. 1 0 3
4. **1 2 3**

13) What is the output of this program?

class multidimention\_array

{

public static void main(String args[])

{

int arr[][] = new int[3][];

arr[0] = new int[1];

arr[1] = new int[2];

arr[2] = new int[3];

int sum = 0;

for (int i = 0; i < 3; ++i)

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

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

for (int i = 0; i < 3; ++i)

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

sum + = arr[i][j];

System.out.print(sum);

}

}

1. 11
2. **10**
3. 13
4. 14

14) What is the output of this program?

class leftshift\_operator

{

public static void main(String args[])

{

byte x = 64;

int i;

byte y;

i = x << 2;

y = (byte) (x << 2)

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

}

}

1. 0 64
2. 64 0
3. 0 256
4. **256 0**

15) What is the output of this program?

class access

{

static int x;

void increment()

{

x++;

}

}

class static\_use

{

public static void main(String args[])

{

access obj1 = new access();

access obj2 = new access();

obj1.x = 0;

obj1.increment();

obj2.increment();

System.out.println(obj1.x + " " + obj2.x);

}

}

1. 1 2
2. 1 1
3. **2 2**
4. Compilation Error

Section: Coding

16) ***Minimum Sum of array***

﻿Given a sorted integer array. We need to make array elements distinct by increasing values and keeping array sum minimum possible.

Complete the *minSum* function in the editor below to create an array of unique elements with a minimal sum. Return the integer sum of the resulting array.

Sample Input 1:

5

2 2 3 5 6

Sample Output 1

20

Explanation: We make the array as {2, 3, 4, 5, 6}.

Sum becomes 2 + 3 + 4 + 5 + 6 = 20

Sample Input 2:

4

3 4 6 8

Sample Output 2

21

Explanation: All elements are unique so result is sum of each elements.

Sample Input 3:

3

1 2 2

Sample Output 3:

6

Explanation: The duplicate array elements 2 must be addressed. The minimum unique array will be achieved by incrementing one of the twos by 1, creating the array [1, 2, 3].

The sum of elements in the new array is 1 + 2 + 3 = 6.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 4  3  4  6  8 | 2  34  34 | 5  2  3  7  7  7 | 4  34  36  39  39 | 7  2  4  7  8  8  9  9 |
| 21 | 69 | 29 | 149 | 51 |

**Solution**:

import java.util.\*;

public class file

{

static int minSum(int arr[])

{

int sum = arr[0];

int n=arr.length;

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

{

if (arr[i] == arr[i - 1]) {

// While current element is same as

// previous or has become smaller

// than previous.

int j = i;

while (j < n && arr[j] <= arr[j - 1])

{

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

j++;

}

}

sum = sum + arr[i];

}

return sum;

}

public static void main(String[] args) {

// TODO code application logic here

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();

System.out.println(minSum(arr));

}

}

17***) Swap and Reverse***

Help Harsh to exchange the first and last word and reverse all other characters of an input string.

If string contains two words then swap them and print as output

Sample Input 1

I am a good coder

Sample Output 1

coder doog a ma I

Sample Input 2

Harsh Singla

Sample Output 2

Singla Harsh

Explanation 1

First and last words are swapped and rest all characters are reversed.

Explanation 2

As there are two words in string we will swap them as output.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Hello World | Chitkara University is the best. | A b C d E F GH . | Print it as such. | Enter Sample input in given area |
| World Hello | best. eht si ytisrevinU Chitkara | . HG F E d C b A | such. sa ti Print | area nevig ni tupni elpmaS Enter |

**Solution:**

import java.util.\*;

public class file{

static void swap\_rev(String s)

{

String fst = "";

int i = 0;

for (i = 0; i < s.length();) {

// Iterating from starting index

// When we get space, loop terminates

while (s.charAt(i) != ' ') {

fst = fst + s.charAt(i);

i++;

}

// After getting one Word

break;

}

// Taking an Empty String

String last = "";

int j = 0;

for (j = s.length() - 1; j >= i;) {

// Iterating from last index

// When we get space, loop terminates

while (s.charAt(j) != ' ') {

last = s.charAt(j) + last;

j--;

}

// After getting one Word

break;

}

// Printing last word

System.out.print(last);

for (int m = j; m >= i; m--) {

// Reversing the left characters

System.out.print(s.charAt(m));

}

// Printing the first word

System.out.println(fst);

}

public static void main(String[] args)

{ Scanner sc=new Scanner(System.in);

String st=sc.nextLine();

swap\_rev(st);

}

}

***18) Continuous Sub Array***

Help Sunil to find all possible Continuous Sub Array in an Array Whose Sum Is Equal to Given Number, k. If no Continuous Sub Array exists then print “No sub array exist” message on screen.

Complete the *findSubArray* function.

Sample Input1

6 // number of element in array

42

15

12

8

6

32

26 //number K

Sample Output1

12 8 6

Sample Input2

5

13

12

3

10

18

28

Sample Output2

13 12 3

10 18

Explanation

The first line of input contains N. N is number of element in array.

Next is the N element new line separated input by user.

The last input is a number K. Continuous Sub Array sum will be equal to K.

Sample Output represents the Space separated Number whose sum is equal to K.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 6  12  5  31  13  21  8  49 | 3  3  4  5  7 | 6  18  4  5  6  8  7  15 | 6  23  12  12  12  24  26  62 | 5  3  5  1  3  4  8 |
| 5 31 13 | 3 4 | 4 5 6  8 7 | 12 24 26 | 3 5  1 3 4 |

**Solution**:

import java.util.\*;

public class file

{

static void findSubArray(int[] inputArray, int inputNumber)

{

int sum = 0;

int flag=0;

//Iterating through 'inputArray'

for (int i = 0; i < inputArray.length; i++)

{

//Assigning inputArray[i] to 'sum'

sum = inputArray[i];

for (int j = i+1; j < inputArray.length; j++)

{

//Adding inputArray[j] to 'sum'

sum = sum + inputArray[j];

//If 'sum' is equal to 'inputNumber' then printing the sub array

if(sum == inputNumber)

{

//System.out.println("Continuous sub array of "+Arrays.toString(inputArray)+" whose sum is "+inputNumber+" is ");

for (int k = i; k <= j; k++)

{

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

flag=1;

}

System.out.println();

}

//if 'sum' is smaller than 'inputNumber', continue the loop

else if (sum < inputNumber)

{

continue;

}

//if 'sum' is greater than 'inputNumber', then break the loop

else if (sum > inputNumber)

{

//System.out.println("No sub array exist");

break;

}

}

}

if(flag==0)

System.out.println("No sub array exist");

} //end of function

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();

int k=sc.nextInt();

findSubArray(arr,k);

}

}