**1.calculateSal**

Read the question carefully and follow the input and output format.  
  
Dinesh got salary for this month and he spends 20% of his salary for food and 30% of his salary for travel. If he takes care of other shifts he will get 2% of the salary per day. Given his salary and the number of shifts he handled. Calculate how much he can save in his pocket after spending all these?  
  
**Input and Output Format :**  
First line of input consists of an integer, salary. Next line correspond to the number of shifts. Output consist of an integer, which is saving.  
  
1) Print "Salary too large" when salary is greater than 8000.  
2) Print "Shifts too small" when the shift is less than 0.  
3) Print "Salary too small" when the salary is less than 0.  
  
Include a function named calculateSal(int salary, int shifts) whose return type is an integer, which is the saving.  
  
**Sample Input 1:**  
7000

5  
  
**Sample Output 1:**  
4200  
  
**Sample Input 2:**  
1000

5  
**Sample Output 2:**  
600

**Sample Input 3:**  
0

5

**Implementation:**

import java.util.Scanner;

public class CalculateSalary

{

public static void main(String[] args) {

int salary=0,shifts=0; double savings=0;

Scanner in=new Scanner(System.in);

salary = in.nextInt();

shifts = in.nextInt();

if(salary > 8000)

System.out.print("Salary too large ");

else if(shifts<=0)

System.out.print ("Shifts too small\n");

else if(salary<=0)

System.out.print ("Salary too small");

else {

savings = ((salary\*0.5)+(salary\*0.02\*shifts));

System.out.printf ("%.0f",savings);

}

}

}

**Output:**

**1)**7000

5

4200

**2)**8001

5

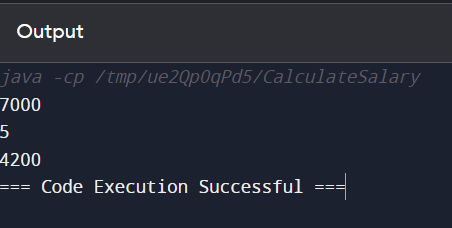
Salary too large

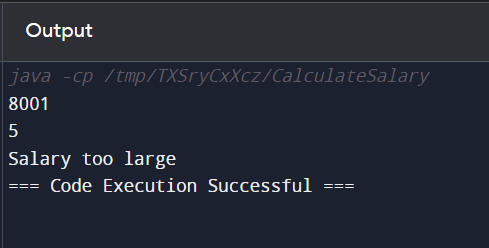
**3)**0

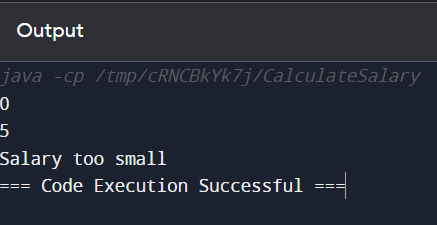
5

Salary too small

**Screenshot:**

****

****

****

**2.Repeated Salary Count**

 John is working as a clerk in an organization where N number of people are working. His boss has asked him to get the count of employees who get same salary. Help him to get the count of repeated salary.

 Include a function named **countRepeaters** that accepts 2 arguments and returns an int. The first argument is the input array and the second argument is an int that corresponds to the size of the array. The function returns an int that corresponds to the number of repeaters.

 If the size of the array is negative or if any of the array elements are negative, print “Invalid Input” and terminate the program.

**Input and Output Format:**

Input consists of n+1 integers. The first integer corresponds to n, the number of elements in the array. The next 'n' integers correspond to the elements in the array.

Output consists of an integer that corresponds to the number of repeaters.

 Assume that utmost one element in the array would repeat.

Assume that the maximum number of elements in the array is 20.

**Sample Input 1:**

5

1000

2000

3500

2000

5000

**Sample Output 1:**

2

**Sample Input 2:**

-5

**Sample Output 2:**

Invalid Input

**Sample Input 3:**

5

1000

-2000

**Sample Output 3:**

Invalid Input

**Implementation:**import java.util.\*;

public class RepeatedSalaryCount {

public static void main(String[] args) {

int n, i,j,k,count=1;

Scanner in=new Scanner(System.in);

n = in.nextInt();

if(n < 0) {

System.out.print("Invalid array size");

System.exit(0);

}

else {

int a[]=new int[100];

for(i = 0; i< n; i++) {

a[i] = in.nextInt();

if(a[i] < 0) {

System.out.print("Invalid input");

System.exit(0);

}

}

for(i=0;i<n;i++) {

for(j=i+1;j<n;) {

if(a[i]==a[j]) {

count++;

for(k=j;k<n;k++)

a[k]=a[k+1]; n--;

}

else

j++;

}

}

System.out.print(count);

}

}

}

**Input 1:**

5

6000

8000

4000

2000

4000

**Output 1:**

2

**Input 2:**

-2

**Output 2:**

Invalid array size

**Input 3:**

5

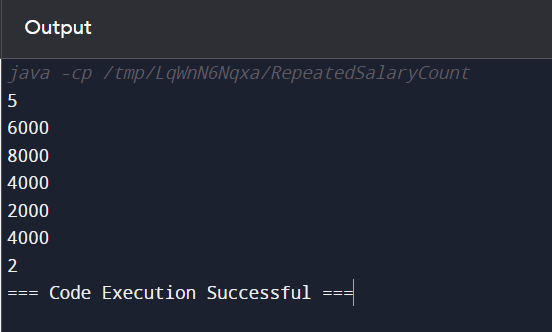
9000

-6000

**Output 3:**

Invalid input

**Screenshot:**

****

**3.findCricketerId**

Read the question carefully and follow the input and output format.  
  
Given an input array first Index indicates the cricketer’s id and second index indicates the score and so on……Find out the cricketer's id who scored more than given score  
  
**Input and Output Format :**  
First line of input consists of n, the number of elements. Next n lines correspond to the array elements.The next line of the input consists of an integer that corresponds to the given score. Output consist of an integer array, which contains cricketer's id who have scored more than the given score.  
  
1) Print "Invalid array size" when size of the array is negative and terminate the program .  
2) Print "Invalid input" when there is any negative numbers available in the input array and terminate the program.  
3) Print "Invalid score" when the score is negative.  
  
Include a function named findCricketerId(int array[], int size, int score) whose return type is void.  
The output array is stored in a global variable named cricketer.  
  
**Sample Input 1:**  
6  
1  
1000  
5  
2000  
3  
4000  
1000  
  
**Sample Output 1:**  
5  
3

**Implementation:**

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

int n, i,j=0,score;

Scanner in=new Scanner(System.in);

n = in.nextInt();

if(n < 0) {

System.out.print("Invalid array size");

System.exit(0);

}

else {

int a[]=new int[n];

for(i = 0; i< n; i++) {

a[i] = in.nextInt();

if(a[i] < 0) {

System.out.print("Invalid input");

System.exit(0);

}

}

score = in.nextInt();

if(score<0) {

System.out.print ("Invalid score");

System.exit(0);

}

int cricketer[]=new int[100];

for(i=1;i<n;i=i+2) {

if(a [i]>score) {

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

j++;

}

}

for(i=0;i<j;i++) {

System.out.println(cricketer[i]);

}

}

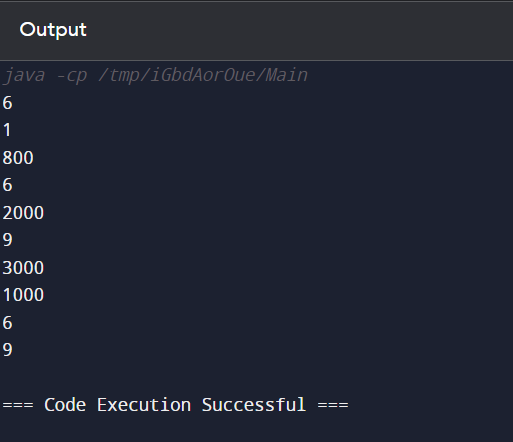
}

}

**Input:**  
6  
1  
800  
6  
2000  
9  
3000  
1000

**Output:**  
6  
9

**Screenshot:**



**4.highestFeedBack**

Read the question carefully and follow the input and output format.  
  
In a company there are some managers working on two different projects (MetLife and Hardfort). When the feedback was taken their feedback was present in both MetLife Feedback as well as Hardfort Feedback. Write a method to create a consolidated feedback for the managers for MetLife and HardForts. For those working on both the projects the highest feedback is taken. In the 2 given arrays, the First Index represents the Employee id and second one Represents The Feed Back Score and so on....  
  
**Input and Output Format:**  
First line corresponds to n, the size of the array. The next n lines correspond to the elements of the first array. The next n lines correspond to the elements in the second array. Output corresponds to the consolidated feedback score.  
  
1) Print "Invalid array size" when size of the array is a negative number and terminate the program  
2) Print "Invalid input" when there is any negative number available in the input array and terminate the program  
  
Include a function named highestFeedBack(int metlife[],int hardfort[],int size) whose return type is void.  
The output array is stored in a global variable named fedback.  
  
**Sample Input 1:**  
8  
1  
90  
2  
75  
3  
92  
5  
85  
1  
80  
2  
85  
3  
80  
4  
85  
**Sample Output 1:**  
1  
90  
2  
85  
3  
92  
5  
85  
4  
85  
  
**Sample Input 2:**  
5  
5  
8  
9  
1  
-6  
**Sample Output 2:**  
Invalid number  
  
**Sample Input 3:**  
-4  
**Sample Output 3:**  
Invalid array size

**Implementation:**

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

int n, i,j,k=0,count,count1;

Scanner in=new Scanner(System.in);

n = in.nextInt();

if(n < 0) {

System.out.print("Invalid array size");

System.exit(0);

} else {

int a[]=new int[n];

for(i = 0; i< n; i++) {

a[i] = in.nextInt();

if(a[i] < 0) {

System.out.print("Invalid input");

System.exit(0);

}

}

int b[]=new int[n];

for(i = 0; i< n; i++) {

b[i] = in.nextInt();

if(b[i] < 0) {

System.out.print("Invalid input");

System.exit(0);

}

}

int c[]=new int[100];

for(i=0;i<n;i=i+2) {

count=0;

for(j=0;j<n;j=j+2) {

if(a[i]==b[j]) {

count=1;

if(a[i+1]>b[j+1]) {

c[k]=a[i];

c[++k]=a[i+1];

k++;

} else {

c[k]=a[i];

c[++k]=b[j+1];

k++;

}

}

}

if(count==0) {

c[k]=a[i];

c[++k]=a[i+1];

k++;

}

}

for(i=0;i<n;i=i+2) {

count1=0;

for(j=0;j<n;j=j+2) {

if(b[i]==a[j]) {

count1=1;

}

}

if(count1!=1) {

c[k]=b[i];

c[++k]=b[i+1];

k++;

}

}

for(i=0;i<k;i++) {

System.out.println(c[i]);

}

}

}

}

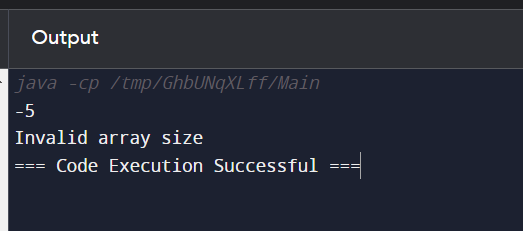
**Input:**

-5

**Output:**

Invalid array size

**Screenshot:**



**5. powerOfTwo**

Read the question carefully and follow the input and output format.  
  
Check whether given number is a power of 2 or not .If yes Print 'Yes' else 'No'  
  
**Input and Output Format :**  
Input consists of an integer number. And output is a single line that displays 'Yes' or 'No'  
  
Print "Number too small" if the number is less than 0  
Print "Number too large" if the number is greater than 32767  
  
Include a function named powerOfTwo(int n) that returns an integer.  
  
  
**Sample Input 1:**  
3  
**Sample Output 1:**  
No  
  
**Sample Input 2**:  
34569  
**Sample Output 2:**  
Number too large

**Implementation:**

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

int number = scanner.nextInt();

powerOfTwo(number);

}

public static void powerOfTwo(int n) {

if (n < 0) {

System.out.println("Number too small");

} else if (n > 32767) {

System.out.println("Number too large");

} else if (isPowerOfTwo(n)) {

System.out.println("Yes");

} else {

System.out.println("No");

}

}

public static boolean isPowerOfTwo(int n) {

if (n == 0) {

return false;

}

while (n % 2 == 0) {

n /= 2;

}

return n == 1;

}

}

**Output:**

**Input 1:**

4

**Output 1:**

Yes

**Input 2:**

9

**Output 2:**

No

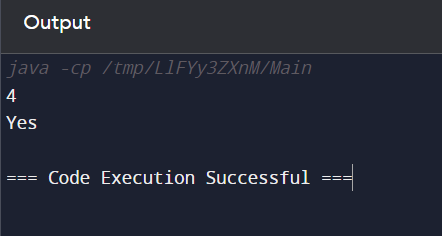
**Input 3:**

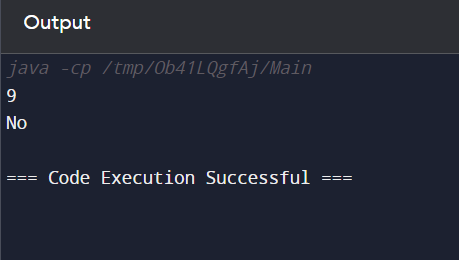
82369

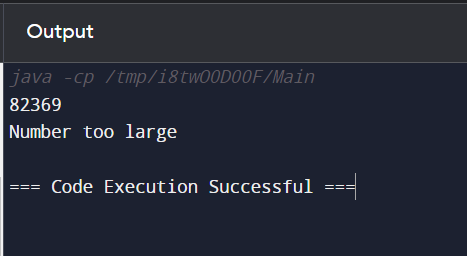
**Output 3:**

Number too large

**Screenshot:**

****

****

****