1. Given a square matrix, calculate the absolute difference between the sums of its diagonals.

For example, the square matrix arr is shown below:

1 2 3

4 5 6

9 8 9

The left-to-right diagonal =1+5+9=15. The right to left diagonal =3+5+9=17. Their absolute difference is|15-17|=2.

**Answer:**

public class Solution {

public static void main(String[] args) {

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

long sumLeftD=0,sumRightD=0; int n=sc.nextInt();

long [][] arr=new long[n][n];

for(int i=0;i<n;i++){for(int j=0;j<n;j++){arr[i][j]=sc.nextLong();

if(i==j){ sumLeftD+=arr[i][j];} if((i+j)==n-1){ sumRightD+=arr[i][j];}

}}

System.out.println(Math.abs(sumLeftD-sumRightD));

}}}

1. You oversee the cake for your niece's birthday and have decided the cake will have one candle for each year of her total age. When she blows out the candles, she’ll only be able to blow out the tallest ones. Your task is to find out how many candles she can successfully blow out.

**Input Format**

The first line contains a single integer, n, denoting the number of candles on the cake.   
The second line contains n space-separated integers, where each integer i describes the height of candle.

Sample Input:

4

3 2 1 3

Output:

2

**Answer:**

public class Solution {

static int birthdayCakeCandles(int n, int[] ar) { int max=ar[0],count=0;

for(int i=0;i<ar.length;i++){ if(max==ar[i]){ count++;}else if(max<ar[i]){max=ar[i]; count=1; }}return count; }

public static void main(String[] args) { Scanner in = new Scanner(System.in);

int n = in.nextInt(); int[] ar = new int[n];

for(int ar\_i = 0; ar\_i < n; ar\_i++){ ar[ar\_i] = in.nextInt(); }

int result = birthdayCakeCandles(n, ar);System.out.println(result);}}

1. Amtrust University has the following grading policy:

* Every student receives a grade in the inclusive range from 0 to 100.
* Any grade less than 40 is a failing grade.

Sam is a professor at the university and likes to round each student's grade according to these rules:

* If the difference between the grade and the next multiple of  5 is less than 3, round  grade up to the next multiple of  5.
* If the value of grade is less than 38, no rounding occurs as the result will still be a failing grade.

**Input Format**

* The first line contains a single integer, n , the number of students.   
  Each line  i of the  n subsequent lines contains a single integer, grades[i] , denoting i student 's grade.

**Sample Input:**

4

73

67

38

33

**Sample Output:**

75

67

40

33

**Answer:**

import java.io.\*;

import java.util.\*;

import java.text.\*;

import java.math.\*;

import java.util.regex.\*;

public class Solution {

static int[] solve(int[] grades) {

if (grades == null || grades.length == 0) {

return grades;

}

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

int remainder = 5-(grades[i] % 5);

if (!(remainder == 0) && remainder < 3 && grades[i]>=38) {

grades[i] = grades[i] + remainder;

}

}

return grades;

}

public static void main(String[] args) {

Scanner in = new Scanner(System.in);

int n = in.nextInt();

int[] grades = new int[n];

for(int grades\_i=0; grades\_i < n; grades\_i++){

grades[grades\_i] = in.nextInt();

}

int[] result = solve(grades);

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

System.out.print(result[i] + (i != result.length - 1 ? "\n" : ""));

}

System.out.println("");

}

}

1. Merge two arrays into one without using third array with Time Complexity as O(n):

void mergeSortedArrays(int arr1[],int arr2[],int m,int n)

{

int i,j,k;

i=m-1;

j=n-1;

k=m+n-1;

while(i>=0&&j>=0)

{

if(arr1[i]>arr2[j])

arr1[k]=arr1[i--];

else

arr1[k]=arr2[j--];

k--;

}

while(j>=0)

arr1[k--]=arr2[j--];

}