

CSE_CS104_Practice1_2019

Test Summary

- No. of Sections: 2
- No. of Questions: 31
- Total Duration: 11000 min

Section 1 - Coding

Section Summary

- No. of Questions: 15
- Duration: 10000 min

Additional Instructions:

None

Q1. Count the frequency of each element of given array

Input Format

```
10  
8 9 1 2 6 7 2 2 6 7
```

Output Format

```
8 occurs 1 times  
9 occurs 1 times  
1 occurs 1 times  
2 occurs 3 times  
6 occurs 2 times  
7 occurs 2 times
```

Sample Input

```
5  
1  
2  
1
```

Sample Output

```
1 occurs 2 times  
2 occurs 2 times  
3 occurs 1 times
```

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

Q2. Given two strings S and A. Print "1" if both strings are anagrams otherwise print "0".

Input Format

First line of input contains a single integer T which denotes the number of test cases. T test cases follows, first line of each test case contains a string S. Second and last line of each test case consists of string A.

```
2
cdbkdub
dsbkcsdn
hello
elhlo
```

Output Format

Corresponding to each test case, print the required output.

```
0
1
```

Constraints

$1 \leq T \leq 100$
 $1 \leq \text{length}(S \text{ and } A) \leq 1000$

Sample Input

```
2
cdbkdub
dsbkcsdn
hello
```

Sample Output

```
0
1
```

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

Q3. Print the sum of array elements

Input Format

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

Output Format

```
15
```

Sample Input

```
5
1
2
3
```

Sample Output

```
15
```

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

Q4. *Special String*

A string is called a special string if it does not contain a vowel i.e., (a,e,i,o,u) in the first three characters or last three characters. Given a set of strings, output the number of special strings in it.

Input Format

The first line of the input contains the number of strings n.

It is followed by n lines of strings, S_i .

Output Format

Output the number of special strings in the given set.

Constraints

$0 < n \leq 100$

$1 \leq \text{length of } S_i \leq 100$

Sample Input

```
3
aetopp
ddfghk
onetwo
```

Sample Output

```
1
```

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

Q5. Reverse String

Given a string, that contains special character together with alphabets ('a' to 'z' and 'A' to 'Z'), reverse the string in a way that special characters are not affected.

Input Format

The first line of the input contains the number of test cases, t.

It is followed by t lines of strings.

Output Format

Output the t reversed strings in a way that special characters are not affected.

Sample Input

```
2
a,b$c
Ab,c,de!$
```

Sample Output

```
c,b$a
ed,c,bA!$
```

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

Q6. Chef likes to write poetry. Today, he has decided to write a X pages long poetry, but unfortunately his notebook has only Y pages left in it. Thus he decided to buy a new CHEFMATE notebook and went to the stationary shop. Shopkeeper showed him some N notebooks, where the number of

pages and price of the i^{th} one are P_i pages and C_i rubles respectively. Chef has spent some money preparing for Ksen's birthday, and then he has only K rubles left for now. Chef wants to buy a single notebook such that the price of the notebook should not exceed his budget and he is able to complete his poetry. Help Chef accomplishing this task. You just need to tell him whether he can buy such a notebook or not. Note that Chef can use all of the Y pages in the current notebook, and Chef can buy only one notebook because Chef doesn't want to use many notebooks.

Input Format

The first line of input contains an integer T , denoting the number of test cases. Then T test cases are follow.

The first line of each test case contains four space-separated integers X, Y, K and N , described in the statement. The i^{th} line of the next N lines contains two space-separated integers P_i and C_i , denoting the number of pages and price of the i^{th} notebook respectively.

Output Format

For each test case, Print "**LuckyChef**" if Chef can select such a notebook, otherwise print "**UnluckyChef**" (quotes for clarity).

Constraints

- $1 \leq T \leq 10^5$
- $1 \leq Y < X \leq 10^3$
- $1 \leq K \leq 10^3$
- $1 \leq N \leq 10^5$
- $1 \leq P_i, C_i \leq 10^3$

Sample Input

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

Sample Output

```
LuckyChef
UnluckyChef
UnluckyChef
```

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

- Q7. Little chief has his own restaurant in the city. There are N workers there. Each worker has his own salary. The salary of the i -th worker equals to W_i ($i = 1, 2, \dots, N$). Once, chief decided to equalize all workers, that is, he wants to make salaries of all workers to be equal. But for this goal he can use only one operation: choose some worker and increase by 1 salary of each worker, except the salary of the chosen worker. In other words, the chosen worker is the loser, who will be the only worker, whose salary will be not increased during this particular operation. But loser-worker can be different for different operations, of course. Chief can use this operation as many times as he wants. But he is a busy man. That's why he wants to minimize the total number of operations needed to equalize all workers. Your task is to find this number.

Input Format

The first line of the input contains an integer T denoting the number of test cases. The description of T test cases follows. The first line of each test case contains a single integer N denoting the number

of workers. The second line contains N space-separated integers W_1, W_2, \dots, W_N denoting the salaries of the workers.

Output Format

For each test case, output a single line containing the minimum number of operations needed to equalize all workers.

Constraints

- $1 \leq T \leq 100$
- $1 \leq N \leq 100$
- $0 \leq W_i \leq 10000 (10^4)$

Sample Input

```
2
3
1 2 3
?
```

Sample Output

```
3
0
```

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

Q8. Forgotten languages (also known as extinct languages) are languages that are no longer in use. Such languages were, probably, widely used before and no one could have ever imagined that they will become extinct at some point. Unfortunately, that is what happened to them. On the happy side of things, a language may be dead, but some of its words may continue to be used in other languages.
Using something called as *the Internet*, you have acquired a dictionary of N words of a forgotten language. Meanwhile, you also know K phrases used in modern languages. For each of the words of the forgotten language, your task is to determine whether the word is still in use in any of these K modern phrases or not.

Input Format

The first line of the input contains an integer T denoting the number of test cases. The description of T test cases follows.

The first line of a test case description contains two space separated positive integers N and K . The second line of the description contains N strings denoting a dictionary of the forgotten language. Each of the next K lines of the description starts with one positive integer L denoting the number of words in the corresponding phrase in modern languages. The integer is followed by L strings (not necessarily distinct) denoting the phrase.

Output Format

For each test case, output a single line containing N tokens (space-separated): if the i^{th} word of the dictionary exists in at least one phrase in modern languages, then you should output **YES** as the i^{th} token, otherwise **NO**

Constraints

- $1 \leq T \leq 20$
- $1 \leq N \leq 100$
- $1 \leq K, L \leq 50$
- $1 \leq \text{length of any string in the input} \leq 5$

Sample Input

```
2  
3 2  
piygu ezyfo rzotm  
1 nivau
```

Sample Output

```
YES YES NO  
NO NO NO YES
```

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

Q9. John Watson knows of an operation called a *right circular rotation* on an array of integers. One rotation operation moves the last array element to the first position and shifts all remaining elements right one. To test Sherlock's abilities, Watson provides Sherlock with an array of integers. Sherlock is to perform the rotation operation a number of times then determine the value of the element at a given position.

For each array, perform a number of right circular rotations and return the value of the element at a given index.

For example, array , $a=[3,4,5]$ number of rotations, $k=2$ and indices to check, $m=[1,2]$.

First we perform the two rotations:

$[3,4,5] \rightarrow [5,3,4] \rightarrow [4,5,3]$

Now return the values from the zero-based indices 1 and 2 as indicated in the m array.

$a[1] = 5$

$a[2] = 3$

Input Format

The first line contains 3 space-separated integers n, k, and q, the number of elements in the integer array, the rotation count and the number of queries.

The second line contains n space-separated integers, where each integer i describes array element $a[i]$ (where $0 \leq i \leq n$).

Each of the q subsequent lines contains a single integer m denoting , the index of the element to return from a.

Output Format

For each query, print the value of the element at index m of the rotated array on a new line.

Constraints

$1 \leq n \leq 10^5$

$1 \leq a[i] \leq 10^5$

$1 \leq k \leq 10^5$

Sample Input

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

Sample Output

```
2  
3  
1
```

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

- Q10. You are given a phone book that consists of people's names and their phone number. After that you will be given some person's name as query. For each query, print the phone number of that person.

Input Format

The first line will have an integer n denoting the number of entries in the phone book. Each entry consists of two lines: a name and the corresponding phone number.

After these, there will be some queries. Each query will contain a person's name. Read the queries until end-of-file.

Constraints:

A person's name consists of only lower-case English letters and it may be in the format 'first-name last-name' or in the format 'first-name'. Each phone number has exactly 8 digits without any leading zeros.

Output Format

For each case, print "Not found" if the person has no entry in the phone book. Otherwise, print the person's name and phone number. See sample output for the exact format.

To make the problem easier, we provided a portion of the code in the editor. You can either complete that code or write completely on your own.

Constraints

$1 \leq n \leq 100000$
 $1 \leq \text{query} \leq 100000$

Sample Input

```
3
gaurav
99999
aman
```

Sample Output

```
aman=55555
Not found
gaurav=99999
```

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

- Q11. **TILING**

This program must calculate how many tiles are needed to tile a floor. The tiles are 8 inches by 8 inches. Tiles can be used as a whole or a part of the tile can be used. Only one usable piece can be cut from a tile. That is, if a piece is cut from a tile, the rest of the tile must be thrown away. The program accepts the length and width of the room and returns how many whole tiles are used and how many part tiles are used. The length is given in inches.

Input Format

Input consists of 2 integers. The first integer corresponds to the width of the room and the second integer corresponds to the length of the room.

Output Format

Output consists of 2 integers. The first integer corresponds to the number of whole tiles used and the second integer corresponds to the number of part tiles used.

Sample Input

```
160  
240
```

Sample Output

```
600  
0
```

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

- Q12. *Lapindrome* is defined as a string which when split in the middle, gives two halves having the same characters and same frequency of each character. If there are odd number of characters in the string, we ignore the middle character and check for lapindrome. For example **gaga** is a lapindrome, since the two halves **ga** and **ga** have the same characters with same frequency. Also, **abccab**, **rotor** and **xyzxy** are a few examples of lapindromes. Note that **abbaab** is NOT a lapindrome. The two halves contain the same characters but their frequencies do not match. Your task is simple. Given a string, you need to tell if it is a lapindrome.

Input Format

First line of input contains a single integer **T**, the number of test cases.

Each test is a single line containing a string **S** composed of only lowercase English alphabet.

```
6  
gaga  
abcde  
rotor  
xyzxy  
abbaab  
ababc
```

Output Format

For each test case, output on a separate line: "YES" if the string is a lapindrome and "NO" if it is not.

```
YES  
NO  
YES  
YES  
NO  
NO
```

Constraints

$$1 \leq T \leq 100$$

$2 \leq |S| \leq 1000$, where $|S|$ denotes the length of **S**

Sample Input

```
6  
gaga  
abcde  
rotor
```

Sample Output

```
YES  
NO  
YES  
YES
```

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

- Q13. A Little Elephant and his friends from the Zoo of Lviv like candies very much. There are N elephants in the Zoo. The elephant with number K ($1 \leq K \leq N$) will be happy if he receives at least A_K candies. There are C candies in all in the Zoo.
- The Zoo staff is interested in knowing whether it is possible to make all the N elephants happy by giving each elephant at least as many candies as he wants, that is, the K^{th} elephant should receive at least A_K candies. Each candy can be given to only one elephant. Print **Yes** if it is possible and **No** otherwise.

Input Format

The first line of the input file contains an integer T , the number of test cases. T test cases follow. Each test case consists of exactly 2 lines. The first line of each test case contains two space separated integers N and C , the total number of elephants and the total number of candies in the Zoo respectively. The second line contains N space separated integers A_1, A_2, \dots, A_N .

Output Format

Output

For each test case output exactly one line containing the string **Yes** if it possible to make all elephants happy and the string **No** otherwise. Output is case sensitive. So **do not print YES or yes**.

Constraints

$1 \leq T \leq 1000$
 $1 \leq N \leq 100$
 $1 \leq C \leq 10^9$
 $1 \leq A_K \leq 10000$, for $K = 1, 2, \dots, N$

Sample Input

```
2
2 3
1 1
2 7
```

Sample Output

```
Yes
No
```

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

- Q14. Here user is asked to enter a one dimensional array and passed to a method `unique_array` and this method finds and removes the numbers of duplicate values in the array. Write a class and method to find duplicate values in array.

Input Format

10 22 10 20 11 22

Output Format

10 22 11 20

Sample Input

Sample Output

6

1 1 2 2 3 3

Enter The Element in the array

Original Array :

1 1 2 2 3 3

Array with unique values.

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

- Q15. Given an array of integers arr[], the task is to count all the pairs (arr[i], arr[j]) such that $i + j = arr[i] + arr[j]$ for all $0 \leq i < j < n$.

Note: Pairs (x, y) and (y, x) are considered a single pair.

Input Format

First Line reads size of array say n
next n lines read n elements of the array

Sample Input

8
0
1
2

Sample Output

1

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

Section 2 - MCQ

Section Summary

- No. of Questions: 16
- Duration: 1000 min

Additional Instructions:

None

- Q1. Which of these class is superclass of String and StringBuffer class?

java.util

java.lang

ArrayList

None

Q2. What is a correct statement structure of java program?

Import, package, interface, class

Package, import, interface, class

Class, import, interface, package

Interface, class, package, import

Q3. Which of the following statements are legal?

- (i) int arr[] [] = new int[5][5];
- (ii) int []arr[] = new int[5][5];
- (iii) int []arr = new int[5][];
- (iv) int []arr = new int[][5];

(i) and (ii)

(iii) and (iv)

(iii) and (i)

(iii) and (ii)

Q4. Observe result of following code:

```
class Test
{
    public static void main(String[] args)
    {
        int x = 10;
        int y = new Test().change(x);
        System.out.println(x+y);
    }
    int change(int x)
    {
        x=12;
    }
}
```

```
    return x;  
}  
}
```

10

24

22

12

Q5. Which of the following is/are true about constructors in Java?

- i) Constructor name should be same as class name.
- ii) If you don't define a constructor for a class, a default parameter-less constructor is automatically created by the compiler.
- iii) The default constructor calls super() and initializes all instance variables to default value like 0, null.
- iv) If we want to parent class constructor, it must be called in first line of constructor.

(i)

(i) , (ii)

(i), (ii), (iii)

(i), (ii), (iii) & (iv)

Q6. 1 What is the output of this program?

```
2  
3 import java.util.*;  
4 class Array  
5 {  
6     public static void main(String args[])  
7     {  
8         int array[] = new int [5];  
9         for (int i = 5; i > 0; i--)  
10            array[5-i] = i;  
11         Arrays.fill(array, 1, 4, 8);
```

```
12     for (int i = 0; i < 5 ; i++)  
13         System.out.print(array[i]);  
14     } |
```

12885

12845

58881

54881

Q7. The smallest Integer type is

byte

short

int

float

Q8. Predict output
double STATIC=3.4;System.out.print(STATIC);

3.4

Error

Raises exception

None

Q9. Modulus operator, %, can be applied to which of these?

Integer

Float

Both Integer and Float

None

Q10. Can 8 byte long data type be automatically type cast to 4 byte float data type?

True

False

Q11. Which of these class is superclass of every class in Java?

string class

Abstract class

object class

ArrayList class

Q12.

```
1 Class Test
2 {
3
4 public static void main(String[] args)
5     {
6         int[] arr[] = new int[3][];
7
8         // Initialize the elements
9         arr[0] = new int[] { 1, 2, 3 };
10        arr[1] = new int[] { 4, 5, 6, 7 };
11        arr[2] = new int[] { 8, 9 };
12    }
```

```
13      // print the array elements
14      for (int[] row : arr)
15          System.out.println(Arrays.toString(row));
16      }
17  }
```

Compile time Error

Run time Error

Display output

[1 2 3]

[4 5 6 7]

[8 9]

None of These

Q13. Which of the following are legal lines of Java code?

1. **int w = (int)888.8;**
2. **byte x = (byte)100L;**
3. **long y = (byte)100;**
4. **byte z = (byte)100L;**

a) 1 and 2

b) 2 and 3

c) 3 and 4

d) All statements are correct

Q14. What is Truncation in Java?

a) Floating-point value assigned to an integer type

b) Integer value assigned to floating type

c) Floating-point value assigned to an Floating type

d) Integer value assigned to floating type

Q15. What will this code print?

```
1 int arr[] = new int [5];  
2 System.out.print(arr);
```

|

a) 0

b) value stored in arr[0].

c) 00000

d) Class name@ hashCode in hexadecimal form

Q16. What is the output of this program?

```
1 class array_output  
2 {  
3     public static void main(String args[])  
4     {  
5         int array_variable [] = new int[10];  
6         for (int i = 0; i < 10; ++i)  
7         {  
8             array_variable[i] = i;  
9             System.out.print(array_variable[i] + " ");  
10            i++;  
11        }  
12    }  
13}  
14
```

|

a) 0 2 4 6 8

b) 1 3 5 7 9

c) 0 1 2 3 4 5 6 7 8 9

d) 1 2 3 4 5 6 7 8 9 10

Answer Key & Solution

Section 1 - Coding

Q1 Test Case

Input

```
6  
87  
85  
87
```

Output

```
87 occurs 3 times  
85 occurs 1 times  
86 occurs 1 times  
89 occurs 1 times
```

Weightage - 40

Input

```
7  
9  
9  
8
```

Output

```
9 occurs 2 times  
8 occurs 2 times  
45 occurs 2 times  
46 occurs 1 times
```

Weightage - 40

Input

```
1  
-1
```

Output

```
-1 occurs 1 times
```

Weightage - 20

Sample Input

```
5  
1  
2  
1
```

Sample Output

```
1 occurs 2 times  
2 occurs 2 times  
3 occurs 1 times
```

Solution

Header

```
import java.util.*;
```

```

public class file {

static void countFreq(int arr[],int n) {
    int freq[]= new int[n];
    for(int i=0;i<n;i++)
        freq[i]=-1;
    for(int i=0;i<n;i++)
    {
        int count=1;
        for(int j=i+1;j<n;j++)
        {
            if(arr[i]==arr[j])
            {
                count++;
                freq[j]=0;
            }
        }
        if(freq[i]!=0)
            freq[i]=count;
    }
    for(int i=0; i<n; i++)
    {
        if(freq[i] != 0)
        {
            System.out.println(arr[i]+ " occurs "+ freq[i]+ " times");
        }
    }
}
}

```

Footer

```

static void main(java.lang.String a[])
{
    Scanner sc=new Scanner(System.in);
    int n=sc.nextInt();
    int arr[]={};
    for(int i=0;i<n;i++)
    {
        arr[i]=sc.nextInt();
    }
    countFreq(arr,n);
}

```

}

Q2 Test Case

Input

2
qwerty
qetywr
world

Output

1
0

Weightage - 20

Input

3
listen
silent
triangle

Output

1
1
1

Weightage - 20

Input

4
listen
listening
zebbee

Output

0
0
1
1

Weightage - 20

Input

5
rail safety
fairy tales
funeral

Output

1
1
0
1

Weightage - 20

Input

Output

6
text
extt
title

1
0
1
1

Weightage - 20

Sample Input

2
cdbkdub
dsbkcsdn
hello

Sample Output

0
1

Solution

Header

```
import java.util.*;  
  
public class file {  
  
    static void sortArray(char ch[], int n)  
    {  
        for(int i=0;i<n-1;i++)  
        {  
            for(int j=0;j<n-i-1;j++)  
            {  
                if(ch[j]>ch[j+1])  
                {  
                    char temp=ch[j];  
                    ch[j]=ch[j+1];  
                    ch[j+1]=temp;  
                }  
            }  
        }  
    }  
    static void checkAnagram(String str1, String str2)  
    {  
        char chr1[],chr2[];  
        int n1=str1.length();  
        int n2=str2.length();
```

```

if(n1!=n2)
{
    System.out.println("0");
}
else
{
    chr1=str1.toCharArray();
    chr2=str2.toCharArray();
    sortArray(chr1, chr1.length);
    sortArray(chr2, chr2.length);

    str1=String.valueOf(chr1);
    str2=String.valueOf(chr2);

    if(str1.equals(str2))
        System.out.println("1");
    else
        System.out.println("0");
}

}

```

Footer

```

public static void main(String args[])
{
    int t;
    String str1,str2;

    Scanner sc=new Scanner(System.in);

    t=sc.nextInt();
    sc.nextLine();
    for(int i=0;i<t;i++)
    {
        str1=sc.nextLine();
        str2=sc.nextLine();
        checkAnagram(str1,str2);
    }
}

```

Q3 Test Case

Input

```
6  
2  
1  
3
```

Output

```
28
```

Weightage - 30

Input

```
4  
2  
1  
4
```

Output

```
10
```

Weightage - 30

Input

```
3  
21  
22  
22
```

Output

```
20
```

Weightage - 40

Sample Input

```
5  
1  
2  
2
```

Sample Output

```
15
```

Solution

```
import java.util.*;  
class ArraySum  
{  
    static void sum(int arr[],int n)  
    {  
        int sum=0;
```

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

Q4 Test Case

Input

2
erty
rty

Output

1

Weightage - 20

Input

3
rtyhgferg
uiklop
eniuuk

Output

1

Weightage - 20

Input

1
ee

Output

0

Weightage - 20

Input

```
4  
ef  
ae  
io
```

Output

```
0
```

Weightage - 20

Input

```
5  
rty  
rfghjkldfg  
escape
```

Output

```
3
```

Weightage - 20

Sample Input

```
3  
aetopp  
ddfghk  
oneatwo
```

Sample Output

```
1
```

Solution**Header**

```
import java.util.Scanner;  
  
public class file {  
  
    static int specialStrng(String strArr[],int n)  
    {  
  
        int res=0;  
        for(int i=0;i<n;i++) {  
            int count=0;  
            for(int j=0;j<strArr[i].length() && j<3;j++) {  
                char ch=strArr[i].charAt(j);  
                if(ch!='a' && ch!='e' && ch!='i' && ch!='o' && ch!='u')  
                    count++;  
            }  
            if(count==3)  
                res++;  
        }  
        return res;  
    }  
}
```

```

        }
        if(count == 3 || count==strArr[i].length())
            res++;
    }
    return res;
}

```

Footer

```

}
public static void main(String args[] ) {
    Scanner s=new Scanner(System.in);
    int n=s.nextInt();
    String strArr[]=new String[n];
    for(int i=0;i<n;i++) {
        strArr[i]=s.next();
    }
    System.out.println(specialStrng(strArr,n));
}
}

```

Q5 Test Case

Input

2
af,jk,,oi:@#g
yuiklo:po;

Output

gi,ok,,jf:@#a
opolki:uy;

Weightage - 20

Input

1
uioy{hello}"World"

Output

dlro{Wolle}"hyoiu"

Weightage - 20

Input

Output

3
r:o:t:o:r
m::o::t::o::r
N----E----W

r:o:t:o:r
r::o:t::o::m
W:::E:::N

Weightage - 20

Input

2
g:
h@

Output

g:
h@

Weightage - 20

Input

3
etr@tre
www
qwerty

Output

ert@rte
www
ytrewq

Weightage - 20

Sample Input

2
a,b\$c
Ab,c,de!\$

Sample Output

c,b\$a
ed,c,bA!\$

Solution

Header

```
import java.util.Scanner;  
public class file  
{  
    static void reverse(char str[])  
    {
```

```

int r = str.length - 1, l = 0;
while (l < r)
{
    if (!Character.isAlphabetic(str[l]))
        l++;
    else if(!Character.isAlphabetic(str[r]))
        r--;
    else
    {
        char tmp = str[l];
        str[l] = str[r];
        str[r] = tmp;
        l++;
        r--;
    }
}

```

Footer

```

}

public static void main(String[] args)
{
    String str;

    Scanner s=new Scanner(System.in);
    int t=s.nextInt();
    for(int i=0;i<t;i++)
    {
        str=s.next();
        char[] charArray = str.toCharArray();
        reverse(charArray);
        String revStr = new String(charArray);
        System.out.println(revStr);
    }
}
}

```

Input

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

Output

```
LuckyChef
UnluckyChef
```

Weightage - 100

Sample Input

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

Sample Output

```
LuckyChef
UnluckyChef
UnluckyChef
```

Solution

```
/* package codechef; // don't place package name! */

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

/* Name of the class has to be "Main" only if the class is public. */
class Codechef
{
    public static void main (String[] args) throws java.lang.Exception
    {
        Scanner sc=new Scanner(System.in);
        int testCases=sc.nextInt();

        for(int i=1;i<=testCases;i++){
            int numberOfPages=sc.nextInt();
            int numberOfPagesAvailable=sc.nextInt();
            int numberOfPagesLeft=numberOfPages-numberOfPagesAvailable;
            int budget=sc.nextInt();
            int numberOfbooks=sc.nextInt();

            int pages,price,flag=0;
            for(int j=1;(j<=numberOfbooks);j++)
            {
```

```

        pages=sc.nextInt();
        price=sc.nextInt();
        if((numberOfPagesLeft<=pages)&&(price<=budget))
        {flag=1;}
    }

    if(flag==1)
        System.out.println("LuckyChef");
    else
        System.out.println("UnluckyChef");
}
}
}

```

Q7 Test Case

Input

1
3
1 2 3

Output

3

Weightage - 100

Sample Input

2
3
1 2 3
2

Sample Output

3
0

Solution

```

/* package codechef; // don't place package name! */

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

/* Name of the class has to be "Main" only if the class is public. */
class codechef

```

```
{  
    static int maxSalary(int arr[])  
{  
        int max=arr[0];  
        for(int j=0;j<arr.length;j++)  
        {  
            if(max<arr[j])  
            {  
                max=arr[j];  
            }  
        }  
        return max;  
    }  
    static int maxIndex(int arr[])  
{  
        int max=arr[0],index=0;  
        for(int j=0;j<arr.length;j++)  
        {  
            if(max<arr[j])  
            {  
                max=arr[j];  
                index=j;  
            }  
        }  
        return index;  
    }  
    static int minSalary(int arr[])  
{  
        int min=arr[0],index=0;  
        for(int j=0;j<arr.length;j++)  
        {  
            if(min>arr[j])  
            {  
                min=arr[j];  
            }  
        }  
        return min;  
    }  
    public static void main (String[] args) throws java.lang.Exception  
{  
        Scanner sc=new Scanner(System.in);  
        int testCases=sc.nextInt();  
  
        for(int i=1;i<=testCases;i++){
```

```

        int numberOfEmployees=sc.nextInt();
        int salaryArray[]=new int[numberOfEmployees];
        int steps=0;
        for(int j=0;j<numberOfEmployees;j++)
        {
            salaryArray[j]=sc.nextInt();

        }
        int diff=0;
        do{
            int min=minSalary(salaryArray);
            int max=maxSalary(salaryArray);
            int index=maxIndex(salaryArray);

            diff=max-min;
            steps+=diff;
            if(diff!=0)
            {
                for(int j=0;(j<numberOfEmployees);j++)
                {
                    if(j!=index)
                        salaryArray[j]+=diff;
                }
            }
        }while(diff!=0);

        System.out.println(steps);

    }
}
}

```

Q8 Test Case

Input

1
4 1
kssdy tjzhy lzym kegqz
1 kegqz kegqz kegqz yyyyi

Output

NO NO NO YES

Weightage - 50

Input

```
1  
3 2  
piygu ezyfo rzotm  
1 piygu
```

Output

```
YES YES NO
```

Weightage - 50

Sample Input

```
2  
3 2  
piygu ezyfo rzotm  
1 piygu
```

Sample Output

```
YES YES NO  
NO NO NO YES
```

Solution

```
import java.util.*;  
import java.lang.*;  
import java.io.*;  
  
/* Name of the class has to be "Main" only if the class is public. */  
class codechef  
{  
  
    public static void main (String[] args) throws java.lang.Exception  
    {  
        Scanner sc=new Scanner(System.in);  
        int testCases=sc.nextInt();  
  
        for(int i=1;i<=testCases;i++){  
            int numberOfforgottenWords=sc.nextInt();  
            String forgottenWordsArray[]=new String[numberOfforgottenWords];  
  
            int forgottenWordsExist[]=new int[numberOfforgottenWords];  
  
            int numberOfModrenSentences=sc.nextInt();  
  
            for(int x=0;x<numberOfforgottenWords;x++)  
            {  
                forgottenWordsArray[x]=sc.next();  
            }  
        }  
    }  
}
```

```

        for(int j=0;j<numberOfModrenSentences;j++)
        {
            int wordsInSenstence=sc.nextInt();
            String toMatch;
            for(int k=0;k<wordsInSenstence;k++)
            { toMatch=sc.next();
                for(int l=0;l<numberOfForgottenWords;l++)
                {
                    if(toMatch.equals(forgottenWordsArray[l]))
                        forgottenWordsExist[l]++;
                }
            }
        }

        for(int x=0;x<numberOfForgottenWords;x++)
        {
            if( forgottenWordsExist[x]>0)
                System.out.print("YES ");
            else
                System.out.print("NO ");
        }

        System.out.println();
    }
}
}

```

Q9 Test Case

Input

4 0 1
5 6 7 8
3

Output

8

Weightage - 20

Input

Output

```
4 4 2  
10 20 30 40  
1  
2
```

```
20  
30
```

Weightage - 30

Input

```
5 1 3  
1 2 3 4 5  
0  
2
```

Output

```
5  
2  
4
```

Weightage - 30

Input

```
1 1 1  
0  
0
```

Output

```
0
```

Weightage - 20

Sample Input

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

Sample Output

```
2  
3  
1
```

Solution

```
import java.util.Scanner;  
  
public class file {  
  
    /**  
     * @param args  
     */  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        int n=sc.nextInt();  
        int k = sc.nextInt();  
        int o = sc.nextInt();  
    }  
}
```

```

    int Q = sc.nextInt();

    int[] array = new int[n];
    for(int i=0;i<n;i++)
    {
        array[(i+k)%n] = sc.nextInt();
    }

    for(int i=0;i<Q;i++)
    {
        System.out.println(array[sc.nextInt()]);
    }
}

```

Q10 Test Case

Input

1
john
444444
john

Output

john=444444

Weightage - 20

Input

5
aaa
111
bbb

Output

aaa=111
Not found
Not found
bbb=111

Weightage - 20

Input

2
a
1
b

Output

a=1
b=2

Weightage - 20

Input

```
3  
aa  
11  
bb
```

Output

```
Not found  
Not found  
Not found  
Not found
```

Weightage - 20

Input

```
5  
mm  
111  
pp
```

Output

```
qq=555  
oo=333  
mm=111
```

Weightage - 20

Sample Input

```
3  
gaurav  
99999  
aman
```

Sample Output

```
aman=55555  
Not found  
gaurav=99999
```

Solution

```
import java.io.*;  
import java.util.*;  
import java.text.*;  
import java.math.*;  
import java.util.regex.*;  
  
public class file {  
  
    public static void main(String[] args) {  
        try {  
            int no_of_entries = 0;  
            int i = 0;  
            String name = null;  
            int number = 0;  
            String query = null;  
  
            HashMap<String, Integer> phoneBook = new HashMap<String, Integer>();  
        } catch (Exception e) {  
            System.out.println("An error occurred: " + e.getMessage());  
        }  
    }  
}
```

```

        BufferedReader b = new BufferedReader(new InputStreamReader(
                System.in));
        no_of_entries = Integer.parseInt(b.readLine());
        while (i < no_of_entries) {
            name = b.readLine();
            number = Integer.parseInt(b.readLine());
            phoneBook.put(name, number);
            i++;
        }
        while (!(query = b.readLine().trim()).isEmpty()) {
            if (phoneBook.containsKey(query))
                System.out.println(query + "=" + phoneBook.get(query));
            else
                System.out.println("Not found");
        }
    } catch (Exception e) {
    }
}
}

```

Q11 Test Case

Input

100
120

Output

180
15

Weightage - 20

Input

100
100

Output

144
25

Weightage - 30

Input

000

Output

10000

ooo
800

ooooo
0

Weightage - 20

Input

90
90

Output

121
23

Weightage - 20

Input

65
60

Output

56
16

Weightage - 10

Sample Input

160
240

Sample Output

600
0

Solution

```
import java.util.*;  
  
public class file {  
  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        int l,b;  
        int tb,t1,w;  
        int el,eb;  
        int p=0;  
        Scanner sc = new Scanner(System.in);
```

```

l =sc.nextInt();
b=sc.nextInt();
tl=(l/8);
tb=(b/8);
w=tl*tb;

el=l-(8*tl);
eb=b-(8*tb);

if(eb>0)
{
    p=tl;
}
if(el>0)
{
    p=p+tb;
}
if(eb>0 && el >0)
{
    p++;
}
System.out.println(w);
System.out.println(p);

}

}

```

Q12 Test Case

Input

1
rotor

Output

YES

Weightage - 100

Sample Input

Sample Output

6
gaga
abcde
rotor

YES
NO
YES
YES

Solution

```
import java.util.Scanner;
import java.util.Arrays;

class Lapin
{
    public static void main(String args[])
    {
        Scanner sc = new Scanner(System.in);
        int t = sc.nextInt();
        while(t!=0)
        {
            String str = sc.next();
            int len = str.length();

            char[] a1 = new char[len/2];
            char[] a2 = new char[len/2];
            if(len%2==0)
            {
                a1 = str.substring(0,len/2).toCharArray();
                a2 = str.substring(len/2,len).toCharArray();
            }
            else
            {
                a1 = str.substring(0,len/2).toCharArray();
                a2 = str.substring((len/2) +1,len).toCharArray();
            }

            Arrays.sort(a1);
            Arrays.sort(a2);
            boolean flag = true;
            for(int i=0;i<len/2;i++)
            {
                if(!(a1[i]==(a2[i])))
                {
                    flag = false;
                    break;
                }
            }
            if(flag)
                System.out.println("YES");
            else
                System.out.println("NO");
        }
    }
}
```

```
        }
        if(flag == false)
            System.out.println("NO");
        else
            System.out.println("YES");
        t--;
    }
}
}
```

Q13 Test Case

Input

```
1
2 3
1 1
```

Output

```
Yes
```

Weightage - 50

Input

```
1
3 7
4 2 2
```

Output

```
No
```

Weightage - 50

Sample Input

```
2
2 3
1 1
3 7
```

Sample Output

```
Yes
No
```

Solution

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

import java.io.*;
```

```

/* Name of the class has to be "Main" only if the class is public. */
class Codechef
{
    public static void main (String[] args) throws java.lang.Exception
    {
        Scanner sc=new Scanner(System.in);
        int testCases=sc.nextInt();

        for(int i=1;i<=testCases;i++){
            int numberOfElephants=sc.nextInt();
            int numberOfCandies=sc.nextInt();
            int sum=0;
            for(int j=1;j<=numberOfElephants;j++)
            {
                sum+=sc.nextInt();
            }

            if(sum<=numberOfCandies)
                System.out.println("Yes");
            else
                System.out.println("No");
        }
    }
}

```

Q14 Test Case

Input

5
2 3 4 3 5

Output

Enter The Element in the array
Original Array :
2 3 4 3 5
Array with unique values :

Weightage - 20

Input

10
23 2 33 33 23 44 33 22 3 12

Output

Enter The Element in the array
Original Array :
23 2 33 33 23 44 33 22 3 12
Array with unique values :

Weightage - 20

Input

```
5  
3 4 3 4 3
```

Output

```
Enter The Element in the array  
Original Array :  
3 4 3 4 3  
Array with unique values :
```

Weightage - 20

Input

```
7  
1 2 3 2 4 5 3
```

Output

```
Enter The Element in the array  
Original Array :  
1 2 3 2 4 5 3  
Array with unique values :
```

Weightage - 20

Input

```
10  
1 2 3 4 1 2 3 4 4 5
```

Output

```
Enter The Element in the array  
Original Array :  
1 2 3 4 1 2 3 4 4 5  
Array with unique values :
```

Weightage - 20

Sample Input

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

Sample Output

```
Enter The Element in the array  
Original Array :  
1 1 2 2 3 3  
Array with unique values :
```

Solution

```
import java.util.*;  
  
public class file {  
    static void unique_array(int[] my_array)  
    {  
        System.out.println("Original Array : ");  
        for (int i = 0; i < my_array.length; i++)  
            System.out.print(my_array[i] + " ");  
        System.out.println();  
        int count = 0;  
        for (int i = 0; i < my_array.length; i++) {  
            for (int j = i + 1; j < my_array.length; j++) {  
                if (my_array[i] == my_array[j])  
                    count++;  
            }  
        }  
        if (count > 0)  
            System.out.println("Array has duplicate values");  
        else  
            System.out.println("Array with unique values");  
    }  
}
```

```
System.out.println("Initial Array : ");

for (int i = 0; i < my_array.length; i++)
{
    System.out.print(my_array[i]+"\t");
}

//Assuming all elements in input array are unique

int no_unique_elements = my_array.length;

//Comparing each element with all other elements

for (int i = 0; i < no_unique_elements; i++)
{
    for (int j = i+1; j < no_unique_elements; j++)
    {
        //If any two elements are found equal

        if(my_array[i] == my_array[j])
        {
            //Replace duplicate element with last unique element

            my_array[j] = my_array[no_unique_elements-1];

            no_unique_elements--;
        }
        j--;
    }
}

//Copying only unique elements of my_array into array1

int[] array1 = Arrays.copyOf(my_array, no_unique_elements);

//Printing arrayWithoutDuplicates

System.out.println();

System.out.println("Array with unique values : ");

for (int i = 0; i < array1.length; i++)
{
    System.out.print(array1[i]+"\t");
}
```

```

        System.out.println("array is " + a);
    }

    System.out.println();
}

public static void main(String[] args)
{
    int n;
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter The Element in the array");
    n=sc.nextInt();
    int a[]=new int[n];

    for(int i=0;i<n;i++)
    {
        a[i]=sc.nextInt();
    }
    unique_array(a);

}
}

```

Q15 Test Case

Input

5
1
2
3

Output

0

Weightage - 20

Input

Output

7
0
1
2

15

Weightage - 20

Input

8
8
7
6

Output

3

Weightage - 20

Input

5
1
1
1

Output

3

Weightage - 20

Input

3
3
2
1

Output

1

Weightage - 20

Sample Input

8
0
1
2

Sample Output

1

Solution

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

```

    static int CountPairs(int arr[], int n)
    {
        int count = 0;

        for (int i = 0; i < n - 1; i++) {
            for (int j = i + 1; j < n; j++) {
                if ((i + j) == (arr[i] + arr[j]))
                    count++;
            }
        }
        return count;
    }

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

            arr[i]=sc.nextInt();
        }
        System.out.print(CountPairs(arr, n));
    }
}

```

Section 2 - MCQ

Q1 java.lang

Solution

No Solution

Q2 Package, import, interface, class

Solution

No Solution

Q3 (i) and (ii)

Solution

No Solution

Q4 22

Solution

No Solution

Q5 (i), (ii), (iii) & (iv)

Solution

No Solution

Q6 58881

Solution

No Solution

Q7 byte

Solution

No Solution

Q8

3.4

Solution

No Solution

Q9 Both Integer and Float

Solution

No Solution

Q10 True

Solution

No Solution

Q11 object class

Solution

No Solution

Q12 Display output

[1 2 3]

[4 5 6 7]

[8 9]

Solution

No Solution

Q13

d) All statements are correct

Solution

No Solution

Q14 a) Floating-point value assigned to an integer type

Solution

No Solution

Q15 d) Class name@ hashCode in hexadecimal form

Solution

No Solution

Q16 a) 0 2 4 6 8

Solution

No Solution