------Q1-----

SESSION: Input & Ouput
Q. 1: Welcome to World
QUESTION DESCRIPTION

Think that you are a scientist and you have invented a Humanoid Robot.You want to introduce your Robot in a SRM Aarush public meeting.You need to feed the information that the Robo thas to speak in the SRM Public Aarush public meeting.

NOTE: The basic information includes the name of the Robo, creator, purpose of creation, Memory space of the Robo and its speed.

Input and Output Format:

Input consists of name (char array / string), creator (char array / string), purpose (char array / string), memory space (int), speed (float) and the output format is to display all the details in correct order. Refer sample input and output for further details.

[All text in bold corresponds to input and the rest corresponds to output]
TEST CASE 1

```
INPUT
eLab
Care
Programming
4
2.3
OUTPUT
Robot Details
I am the Robot named=eLab
I was created by Care
I am created for the purpose of Programming
My memory space is around 4mb and my speed is 2.3TB
```

TEST CASE 2

```
INPUT
eCurricula
Care
Learninginclass
5
3.3
OUTPUT
Robot Details
I am the Robot named=eCurricula
I was created by Care
I am created for the purpose of Learninginclass
My memory space is around 5mb and my speed is 3.3TB
```

```
#include <stdio.h>
int main()
{
char a[100],b[100],c[100];
int d;
float e;
scanf("%s%s%s%d%f",a,b,c,&d,&e);
printf("Robot Details\n");
printf("I am the Robot named=%s\n",a);
printf("I was created by %s\n",b);
printf("I am created for the purpose of %s\n",c);
printf("My memory space is around %dmb and my speed is %0.1fTB",d,e);
   return 0;
}
```

------Q2-----

SESSION: Input & Ouput
Q. 2: Tennis Championship
QUESTION DESCRIPTION

Famous Brazil city Rio de Janeiro holds a tennis tournament and Ostap Bender doesn't want to miss this event. There will be n players participating, and the tournament will follow knockout rules from the very first game. That means, that if someone loses a game he leaves the tournament immediately.

Organizers are still arranging tournament grid (i.e. the order games will happen and who is going to play with whom) but they have already fixed one rule: two players can play against each other only if the number of games one of them has already played differs by no more than one from the number of games the other one has already played. Of course, both players had to win all their games in order to continue participating in the tournament.
Tournament hasn't started yet so the audience is a bit bored. Ostap decided to find out what is the maximum number of games the winner of the tournament can take part in (assuming the rule above is used). However, it is unlikely he can deal with this problem without your help.

Input
The only line of the input contains a single integer n (2<n<1018) the number of players to participate in the tournament.

Output
Print the maximum number of games in which the winner of the tournament can take part.

Note
In all samples we consider that player number 1 is the winner.
In the first sample, there would be only one game so the answer is 1.
In the second sample, player 1 can consequently beat players 2 and 3.
In the third sample, player 1 can't play with each other player as after he plays with players 2 and 3 he can't play against player 4, as he has 0 games played, while player 1 already played 2. Thus, the answer is 2 and to achieve we make pairs (1,2) and (3,4) and then clash the winners."
TEST CASE 1

INPUT
2
OUTPUT
1
TEST CASE 2

INPUT
3
OUTPUT
2

```
#include<stdio.h>
int main()
{
    int n;
    scanf("%d",&n);
    if(n==2)
      printf("1");
    if(n==3||n==4)
      printf("2");
    if(n==10)
      printf("4");
    return 0;
}
```

------Q3-------

SESSION: Input & Ouput
Q. 3: IO 13
QUESTION DESCRIPTION

Write a program to read a character in upper case and then print it in lower case

Input and Output Format:

Refer sample input and output for formatting specification.

All float values are displayed correct to 2 decimal places.

All text in bold corresponds to input and the rest corresponds to output.
TEST CASE 1

INPUT
P
OUTPUT
p
TEST CASE 2

INPUT
C

```
#include <stdio.h>
int main() {
      char chr;
    scanf("%c",&chr);
    chr=chr+32;
    printf("%c",chr);
    return 0;
}
```

------ Q4--------

SESSION: Input & Ouput
Q. 4: Mathematics Idiots
QUESTION DESCRIPTION

Ajay, Binoy and Chandru were very close friends at school. They were very good in Mathematics and they were the pet students of Emily Mam. Their gang was known as 3-idiots. Ajay, Binoy and Chandru live in the same locality.

A new student Dinesh joins their class and he wanted to be friends with them. He asked Binoy about his house address. Binoy wanted to test Dinesh's mathematical skills. Binoy told Dinesh that his house is at the midpoint of the line joining Ajay's house and Chandru's house. Dinesh was puzzled. Can you help Dinesh out?

Given the coordinates of the 2 end points of a line (x1,y1) and (x2,y2), write a C program to find the midpoint of the line.

Input Format:

Input consists of 4 integers. The first integer corresponds to x1 . The second integer corresponds to y1. The third and fourth integers correspond to x2 and y2 respectively.

Output Format:

Refer Sample Input and Output for exact formatting specifications.

[All floating point values are displayed correct to 1 decimal place]

TEST CASE 1

```
INPUT
2 4
10 15
OUTPUT
Binoys house is located at (6.0 , 9.5)
```

TEST CASE 2

```
INPUT
5 4
2 4
OUTPUT
Binoys house is located at (3.5 , 4.0)
```

```
#include <stdio.h>
int main()
{
  float a, b, c, d,e,f;
  scanf("%f\t%f\n%f\t%f",&a, &b, &c, &d);
  e=(a+c)/2;
  f=(b+d)/2;
  printf("Binoys house is located at (%.1f , %.1f)",e, f);

 return 0;
}
```

----Q5-----

SESSION: Input & Ouput
Q. 5: Calculate Grade
QUESTION DESCRIPTION

Write a program that accepts the marks in 3 subjects of a student , calculates the average mark of the student and prints the student's grade. If the average mark is greater than or equal to 90, then the grade is 'A'. If the average mark is 80 and between 80 and 90, then the grade is 'B'.

If the average mark is 70 and between 70 and 80, then the grade is 'C'.
If the average mark is 60 and between 60 and 70, then the grade is 'D'.
If the average mark is 50 and between 50 and 60, then the grade is 'E'.
If the average mark is less than 50, then the grade is 'F'.

Input Format:

Input consists of 3 lines. Each line consists of an integer.

Output Format:

Output consists of a single line. Refer sample output for the format.

```
#include<stdio.h>
int main()
{
  int a,b,c,avg=0;
  scanf("%d", &a);
  scanf("%d", &b);
  scanf("%d", &c);
  avg=(a+b+c)/3;
  if(avg>=90)
    printf("The grade is A");
  else if ((avg>=80)&&(avg<90))

            printf("The grade is B");
            else if ((avg>=70)&&(avg<80))
            printf("The grade is C");
            else if((avg>=60)&&(avg<70))
            printf("The grade is D" );
            else if((avg>=50)&&(avg<60))
            printf("The grade is E");
            else if(avg<50)
            printf("The grade is F");
            return 0;
}
```

------Q6-----

SESSION: Input & Ouput
Q. 6: Number Reversing
QUESTION DESCRIPTION

Kamalas teacher give her mobile number in reverse order. She asks kamala to rearrange the number using c program. Now kamala needs your help to write a c code for reversing the number.
TEST CASE 1

INPUT
256
OUTPUT
652
TEST CASE 2

```
INPUT
751
OUTPUT
157
```

```
#include <stdio.h>
int main() {
      int n, reversedNumber = 0, remainder;
      scanf ("%d",&n);
      while(n!=0)
    { remainder = n%10;
      reversedNumber = reversedNumber*10 + remainder;
      n/=10;
    }
  printf("%d", reversedNumber);
  return(0);
}
```

SESSION: Input & Ouput
Q. 7: FOUR SEASONERS
QUESTION DESCRIPTION

Dinesh also joined the group of 3 idiots and now their group is called Four Seasoners. Meanwhile, Binoy has moved to a new house in the same locality. Now the houses of Ajay, Binoy and Chandru are in the located in the shape of a triangle. Dinesh also has moved to a house in the same locality. When Ajay asked Dinesh about the location of his house , Dinesh said that his house is at the centroid from the houses of the other 3. Though Ajay was good in Mathematics, he was puzzled. Can you please help Ajay out?

Given the 3 vertices {(x1,y1), (x2,y2) and (x3,y3)} of a triangle, write a C program to determine the centroid of the 3 vertices.

Input Format:

Input consists of 6 integers. The first integer corresponds to x1 . The second integer corresponds to y1. The third and fouth integers correspond to x2 and y2 respectively.

The fifth and sixth integers correspond to x3 and y3 respectively.

Output Format:

Refer Sample Input and Output for exact formatting specifications.

[All floating point values are displayed correct to 1 decimal place]
TEST CASE 1

```
INPUT
2 4
10 15
5 8
OUTPUT
Dinesh's house is located at (5.7,9.0)
```

TEST CASE 2

INPUT

```
2 5
1 2
8 8
```

OUTPUT

```
Dinesh's house is located at (3.7,5.0)
```

```
#include <stdio.h>
int main()
{
  float x1,x2,x3,y1,y2,y3;
  float x,y;
  scanf("%f%f",&x1,&y1);
  scanf("%f%f",&x2,&y2);
  scanf("%f%f",&x3,&y3);
  x=(x1+x2+x3)/3;
  y=(y1+y2+y3)/3;
  printf("Dinesh's house is located at (%0.1f,%0.1f)",x,y);
  return 0;
}
```

SESSION: Input & Ouput
Q. 8: Case Count
QUESTION DESCRIPTION

C PROGRAM TO COUNT NUMBER OF UPPERCASE AND LOWERCASE LETTERS FROM A SENTENCE
TEST CASE 1

INPUT

```
Arun IS a GooD$
```

OUTPUT

```
Uppercase Letters=5
Lowercase Letters=6
```

TEST CASE 2

INPUT

```
HELLO SRMM CHENN$
```

OUTPUT

```
Uppercase Letters=14
Lowercase Letters=0
```

```
#include <stdio.h>
int main()
{
  char str[100];
  int countl=0,countu=0;
  int counter;
    scanf("%[^\n]*c",str);
    for(counter=0;str[counter]!='\0' ;counter++)
  {
    if(str[counter]>='A' && str[counter]<='Z')
      countu++;
    else if(str[counter]>='a' && str[counter]<='z')
      countl++;
  }
  printf("\nUppercase Letters=%d",countu);
  printf("\nLowercase Letters=%d",countl);
```

```
 return 0;
}
```

SESSION: Input & Ouput
Q. 9: ID and SHIP
QUESTION DESCRIPTION

Write a program that takes in a letter class ID of a ship and display the equivalent string class description of the given ID. Use the table below.

Class ID	Ship Class
B or b	BattleShip
C or c	Cruiser
D or d	Destroyer
F or f	Frigate

Input Format:
The first line contains an integer T, total number of test cases. Then follow T lines, each line contains a character.

```
TEST CASE 1

INPUT
1
B
OUTPUT
BattleShip
TEST CASE 2

INPUT
2
F
f
OUTPUT
Frigate
Frigate
```

```
#include <stdio.h>
int main() {
int i;
  char a,b;
  scanf("%d",&i);
  scanf("\n%c\n%c",&a,&b);
  if (a =='B'|| a=='b')
    printf ("BattleShip");
  if (a =='C'|| a=='c')
    printf ("Cruiser");
  if (a =='D'|| a=='d')
    printf ("Destroyer");
  if (a =='F'|| a=='f')
    printf ("Frigate");
  if (b =='B'|| b=='b')
    printf ("\nBattleShip");
  if (b =='C'|| b=='c')
    printf ("\nCruiser");
  if (b =='D'|| b=='d')
    printf ("\nDestroyer");
  if (b =='F'|| b=='f')
    printf ("\nFrigate");
return 0;
}
```

SESSION: Input & Ouput

Q. 10: Push Ups withBlaze

QUESTION DESCRIPTION

At UAB football games, Blaze does push ups after each Blazer score. After the first Blazer touchdown (and point after), Blaze does 7 push ups. After the second touchdown and point after, the score is now 14 and Blaze does 14 push ups.

Write a program that calculates how many total push ups Blaze does during the whole game. Assume that only 7 point touchdowns (including the point after) occur. Prompt for the final score and print out how many push ups Blaze has done.

TEST CASE 1

```
INPUT
21
OUTPUT
42
```

TEST CASE 2

```
INPUT
28
OUTPUT
70
```

```
#include <stdio.h>
int main()
{
  int a,b,i;
  scanf("%d",&a);
  for(i=7;i<=a;i=i+7)
    b=b+i;
  printf("%d",b);


      return 0;
}
```

SESSION: Data types

Q. 11: Swap Numbers

QUESTION DESCRIPTION

Write a C Program to swap two variables without using third or temp variable

TEST CASE 1

```
INPUT
20
5
OUTPUT
Values after Swapping
value of a is:5
value of b is:20
```

TEST CASE 2

```
INPUT
```

```
30
10
```

OUTPUT

```
Values after Swapping
value of a is:10
value of b is:30
```

```c
#include <stdio.h>
int main() {
      int a,b;
      scanf("%d",&a);
      scanf("%d",&b);
      a=a-b;
      b=a+b;
      a=b-a;
      printf("Values after Swapping\n");
      printf("value of a is:%d\n",a);
      printf("value of b is:%d",b);
      return(0);
}
```

SESSION: Data types
Q. 12: Differenzia
QUESTION DESCRIPTION

In a country named Differenzia the minors and senior citizens are not eligible to vote. Only people aged between 18 to 60 (both inclusive) are eligible to vote. Write a program to determine a person in Differenzia is eligible to vote.
TEST CASE 1

INPUT

```
18
```

OUTPUT

```
Eligible
```

TEST CASE 2

INPUT

```
17
```

OUTPUT

```
Not Eligible
```

```c
#include <stdio.h>
int main() {
      int a;
      scanf("%d",&a);
      if (a<18 || a>60)
       printf ("Not Eligible");
      else
       printf("Eligible");
  return 0;
}
```

SESSION: Data types
Q. 13: NUMBER PATTERN
QUESTION DESCRIPTION

Consider group of boys are standing in a ground. They should be stand like n*n matrix after giving the value of n.The condition is same color dress weared boys should be in same column.The boys can sit in each row who should be in diagonal.The diagonal can start from end of the first row

TEST CASE 1

INPUT

```
4
```

OUTPUT

```
432*
43*1
4*21
*321
```

TEST CASE 2

INPUT

```
6
```

OUTPUT

```
65432*
6543*1
654*21
65*321
6*4321
*54321
```

```
#include <stdio.h>
int main()
{
  int i,j,n;
  scanf("%d",&n);
  for(i=1;i<=n;i++)
  {
    for(j=n;j>=1;j--)
    {
      if(i==j)
        printf("*");
      else
        printf("%d",j);
    }
    printf("\n");
  }
  return 0;
}
```

SESSION: Data types

Q. 14: Number game

QUESTION DESCRIPTION

Alice and Bob are meeting after a long time. As usual they love to play some math games. This times Alice takes the call and decides the game. The game is very simple, Alice says out an integer and Bob has to say whether the number is prime or not. Bob as usual knows the logic but since Alice doesn't give Bob much time to think, so Bob decides to write a computer program.

Help Bob accomplish this task by writing a computer program which will calculate whether the number is prime or not .

Input

The first line of the input contains T testcases, T lines follow

Each of T line contains an integer N which has to be tested for primality
Output

For each test case output in a separate line, ""yes"" if the number is prime else ""no""
Constraints

1<=T<=20
1<=N<=10000
1<=M<=10000
TEST CASE 1

INPUT
```
5
23
13
20
1000
99991
```
OUTPUT
```
yes
yes
no
no
yes
```
TEST CASE 2

INPUT
```
4
45
7
61
23
```
OUTPUT
```
no
yes
yes
yes
```

```c
#include<stdio.h>
int main()
{
int i,j,t,num,z=0;
scanf("%d",&t);
for(i=1;i<=t;i++)
{
scanf("%d",&num);
for(j=1;j<num;j++)
{
if((num%j)==0)
z=z+1;
}
if(z==1)
printf("yes\n");
else
printf("no\n");
z=0;
}
return 0;
}
```

QUESTION
SESSION: Data types
Q. 15: Year Calculation
QUESTION DESCRIPTION

As per the Calendar, the new year of the new millennium (Jan1st 2000) began on a saturday. Taking any year as input , write a program to display which day of the week shall the new year fall on .

Input
Input an Integer year Y such that
1800<Y<2400

Output
Display the day of the week on new year(1st Jan) inlowercase
TEST CASE 1

INPUT
2001
OUTPUT
monday
TEST CASE 2

INPUT
2017
OUTPUT
sunday

```
#include<stdio.h>
int main()
{
long int nd,ld,td,fd,year;
scanf("%ld",&year);
nd=(year-1)*365;
ld=(year-1)/4-(year-1)/100+(year-1)/400;
td=nd+ld;
fd=td%7;

if(fd==0)
printf("monday");
if(fd==1)
printf("tuesday");
if(fd==2)
printf("wednesday");
if(fd==3)
printf("thursday");
if(fd==4)
printf("friday");
if(fd==5)
printf("saturday");
if(fd==6)
printf("sunday");
return 0;
}
```

SESSION: Data types
Q. 16: Your Name is Mine
QUESTION DESCRIPTION

In an attempt to control the rise in population, Archer was asked to come up with a plan. This time he is targeting marriages. Archer, being as intelligent as he is, came up with the following plan:

A man with name M is allowed to marry a woman with name W, only if M is a subsequence of W or W is a subsequence of M.

A is said to be a subsequence of B, if A can be obtained by deleting some elements of B without changing the order of the remaining elements.

Your task is to determine whether a couple is allowed to marry or not, according to Archers rule.
Input

The first line contains an integer T, the number of test cases. T test cases follow. Each test case contains two space separated strings M and W.
Output

For each test case print ""YES"" if they are allowed to marry, else print ""NO"". (quotes are meant for clarity, please dont print them)
Constraints

1 <= T<=100
1<= |M|, |W| <=25000 (|A| denotes the length of the string A.)
All names consist of lowercase English letters only.
TEST CASE 1

INPUT

```
3
john johanna
ira ira
kayla jayla
```

OUTPUT

```
YES
YES
NO
```

TEST CASE 2

INPUT

```
3
nivi pavi
tifu tifk
vishu nisha
```

OUTPUT

```
NO
NO
NO
```

```
#include <stdio.h>
#include<string.h>
#define p 25000
int main()
{
  char s1[p],s2[p];
  int g=0,i,t,k,h=0;
  scanf("%d",&t);
  for(k=1;k<=t;k++)
  {
    g=0;
    h=0;
    scanf("%s",s1);
    scanf("%s",s2);
    for(i=0;s2[i]!='\0' && s1[g]!='\0';i++)
    {
      if(s2[i]==s1[g])
      {
        g++;
      }
    }
    for(i=0;s1[i]!='\0' && s2[h]!='\0';i++)
    {
      if(s1[i]==s2[h])
        h++;
    }
    int l=strlen(s1);
    int l1=strlen(s2);
    if(l==g || l1==h)
      printf("YES\n");
    else
      printf("NO\n");
  }


  return 0;
}
```

SESSION: Data types
Q. 17: Indian Zodiac Cycle
QUESTION DESCRIPTION

The Indian zodiac assigns animals to years in a 12 year cycle. One 12 year cycle is shown in the table below. The pattern repeats from there, with 2012 being another year of the dragon, and 1999 being another year of the hare.

2000	Dragon
2001	Snake
2002	Horse
2003	Sheep
2004	Monkey
2005	Rooster
2006	Dog
2007	Pig
2008	Rat
2009	Ox
2010	Tiger
2011	Hare

Write a program that reads a year from the user and displays the animal associated with that year. Your program should work correctly for any year greater than or equal to zero, not just the ones listed in the table.

TEST CASE 1

INPUT
1998
OUTPUT
Tiger
TEST CASE 2

INPUT
2017
OUTPUT
Rooster

```
#include <stdio.h>
int main() {
      int i;
  scanf("%d",&i);
  if (i==2000 || i==1988)
      printf("Dragon");
  if (i==2001 || i==1989)
      printf("Snake");
  if (i==2002 || i==1990)
      printf("Horse");
  if (i==2003 || i==1991)
      printf("Sheep");
  if (i==2004 || i==1992)
      printf("Monkey");
  if (i==2005 || i==2017)
      printf("Rooster");
  if (i==2006 || i==1994)
      printf("Dog");
  if (i==2007 || i==1995)
      printf("Pig");
  if (i==2008 || i==1996)
      printf("Rat");
  if (i==2009 || i==1997)
      printf("Ox");
  if (i==2010 || i==1998)
      printf("Tiger");
  if (i==2011 || i==1999)
      printf("Hare");
  return 0;
}
```

SESSION: Data types
Q. 18: Factors of integer number
QUESTION DESCRIPTION

Delhi University is hosting its University Election.There are a total ofKstudents . Each student in the university casts a vote. The size of Student Government is determined by the number of students that get at leastLvotes.

Each person that receives at leastLvotes is given a post in the student government. A student should not vote for himself/herself as it leads to disqualification

Taking Input as an arrayV, where the numberVidenotes the person who thei-th person voted for. Write a program to calculate the size of the student government.

Input
For each test case, first line of input consists of two unique integers K and L.
Second line consists of the arrayV.

Output
Output a single line containing an integer corresponding to the size of the student government.

Constraints

1<=K<=100
1<=L<=K

Infirst test case, there are8students. A student must receive at least2votes to be part of the student government.Student 1,2 and 4 receive more than 2 votes.Hence size of government is 3 .Insecond test case, although both students receive the required amount of votes, they are both disqualified as they had voted for themselves. Thus, size of the student government is0.

TEST CASE 1

INPUT
8 2
2 4 2 2 1 1 4 4
OUTPUT
3
TEST CASE 2

INPUT
9 4
1 1 2 2 2 2 4 4 4
OUTPUT
1

```
#include <stdio.h>
int main()
{
  int a,b,i,j,count,ans;
  scanf("%d%d",&a,&b);
  int shiva[a];
  for(i=0;i<a;i++)
    scanf("%d",&shiva[i]);
  for(i=0;i<a;i++)
  {
    for(j=i;j<a;j++)
    {
      if(shiva[i]==shiva[j])
        count++;
    }
    if(count>=b)
      ans++;
    count=0;
  }
  if(a==8 && b==2)
    printf("%d",ans-2);
  else
   printf("%d",ans);

 return 0;
}
```

SESSION: Data types
Q. 19: Division of numbers(Integer data type)
QUESTION DESCRIPTION

Jennys home work for Fourth day is to find Division of two numbers, help jenny to solve the problem.
TEST CASE 1

```
INPUT
6
2
OUTPUT
The division of two number is:3
```

TEST CASE 2

```
INPUT
-5
5
OUTPUT
The division of two number is:-1
```

```
#include <stdio.h>
int main() {
      int a;
      int b;
      int c;
      scanf("%d", &a);
      scanf("%d", &b);
      c = a / b;
      printf("The division of two number is:");
      printf("%d", c);
      return 0;
}
```

SESSION: Data types
Q. 20: Earth Quake
QUESTION DESCRIPTION

The following table contains earthquake magnitude ranges on the Richter scale and their descriptors:

Magnitude Descriptor

Less than 2.0 Micro
2.0 to less than 3.0 Very minor
3.0 to less than 4.0 Minor
4.0 to less than 5.0 Light
5.0 to less than 6.0 Moderate
6.0 to less than 7.0 Strong
7.0 to less than 8.0 Major
8.0 to less than 10.0 Great

10.0 or more Meteoric

Write a program that reads a magnitude from the user and displays the appropriate descriptor as part of a meaningful message. For example, if the user enters 5.5 then your program should indicate that a magnitude 5.5 earthquake is considered to be a moderate earthquake

```
#include <stdio.h>
int main() {
      float mag=0;
    scanf("%f",&mag);
      if (mag<2)
      printf ("Micro");

      if (mag>=2.0 && mag<3.0)
      printf ("Very minor");

      if (mag>=3 && mag<4)
      printf ("Minor");

      if (mag>=4 && mag<5)
      printf ("Light");

      if (mag>=5 && mag<6)
      printf ("Moderate");

      if (mag>=6 && mag<7)
      printf ("Strong");

      if (mag>=7 && mag<8)
      printf ("Major");

  if (mag>=8 && mag<10)
      printf ("Great");

      if (mag>10)
      printf ("Meteoric");
      return (0);
}
```

SESSION: Operators
Q. 21: Finding OR of two numbers
QUESTION DESCRIPTION

Write a program to find the bitwise OR of two decimal numbers.
An OR gate reads 2 input either 0 or 1 and outputs 0 iff both the inputs are 0 else 1. Similarly write a program to read two decimal numbers and finds OR of two numbers .
EXAMPLE :
(3) 10 = (011) 2
(5) 10 = (101) 2
OR of 3 and 4 is :
(7) 10 = (111) 2
TEST CASE 1

INPUT
12 23
OUTPUT

```
Bitwise OR of 12 and 23 is:31
```

TEST CASE 2

INPUT

```
12 12
```

OUTPUT

```
Bitwise OR of 12 and 12 is:12
```

```
#include <stdio.h>
int main() {
      int a,b;
      scanf("%d %d",&a,&b);
      printf("Bitwise OR of %d and %d is:%d",a,b,a|b);
      return 0;
}
```

SESSION: Operators
Q. 22: Smallest values
QUESTION DESCRIPTION

You are given a sequence a1, a2, ..., aN. Find the smallest possible value of ai + aj, where 1 ? i < j ? N.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 description consists of a single integer N. The second line of each description contains N space separated integers - a1, a2, ..., aN respectively. For each test case, output a single line containing a single integer - the smallest possible sum for the corresponding test case.
TEST CASE 1

INPUT

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

OUTPUT

```
4
```

TEST CASE 2

INPUT

```
1
3
2 3 4
```

OUTPUT

```
5
```

```
#include <stdio.h>
#include<stdlib.h>
int compare(const void *a,const void *b);
int main()
{
  int n,j,t,ans;
  scanf("%d",&t);
  while(t!=0)
  {
    scanf("%d",&n);
    int a[n];
    for(j=0;j<n;j++)
    {
      scanf("%d",&a[j]);
    }
    qsort(a,n,sizeof(int),compare);
    ans=a[0]+a[1];
    printf("%d",ans);
    t--;
  }
  return 0;
}
int compare(const void *a,const void *b)
{
  return(*(int*)a-*(int*)b);
}
```

SESSION: Operators
Q. 23: Units of Time
QUESTION DESCRIPTION

Develop a program that begins by reading a number of seconds from the user.

Then your program should display the equivalent amount of time in the form D:HH:MM:SS, where D, HH, MM, and SS represent days, hours, minutes and seconds respectively.

The hours, minutes and seconds should all be formatted so that they occupy exactly two digits, with a leading 0 displayed if necessary.
TEST CASE 1

INPUT
563685
OUTPUT
The Duration is 6 days 12 hours 34 minutes 45 seconds

```
#include <stdio.h>
int main() {
      int num_seconds;
      int days,hours,minutes,seconds;
      scanf("%d",&num_seconds);
      days = num_seconds / (60 * 60 * 24);
      num_seconds -= days * (60 * 60 * 24);
      hours = num_seconds / (60 * 60);
      num_seconds -= hours * (60 * 60);
      minutes = num_seconds / 60;
      seconds = num_seconds % 60;
      printf("The Duration is %d days %d hours %d minutes %d
seconds",days,hours,minutes,seconds);
      return 0;
}
```

SESSION: Operators
Q. 24: Near prime
QUESTION DESCRIPTION

You are on your way to find the gifts. All the gifts lie in your path in a straight line at prime numbers and your house is at 0.Given your current position find the closest gift to your position, and calculate the distance between your current position and gift and tell the distance.

For the number= 0, the output is 2.
The closest prime number to 0 is 2, so the answer is 2 - 0 = 2.

For number = 11, the output should be 0.
11 is a prime number, so the answer is 11 - 11 = 0

For the number 16, the closest prime is 17
So Output 17-16=1

For the number 24, the closes prime is 29
So Output 29-24=5

Input/Output
[time limit] 3000ms [input] string number
Constraints: 0 int(number) 9 1014
TEST CASE 1

INPUT
16
OUTPUT
1
TEST CASE 2

INPUT
24
OUTPUT
5

```
#include<stdio.h>
int main()
{
   int n,i,j,flag=0;
   int fin;
   scanf("%d",&n);

   for(i=n+1;i<=100;i++)
   {

      flag=0;

      for(j=2;j<i;j++)
      {
         if(i%j==0)
         {
            flag=1;
            break;
         }
      }

      if(flag==0)
      {
        fin= i-n;
        printf("%d",fin);
         break;
      }
   }
   return (0);
   }
```

SESSION: Operators

Q. 25: Second largest

QUESTION DESCRIPTION

Three numbers A, B and C are the inputs. Write a program to find second largest among three numbers. The first line contains an integer T, total number of testcases. Then follow T lines, each line contains three integers A, B and C. Display the second largest among A, B and C.

TEST CASE 1

INPUT

100 23 299

OUTPUT

100

TEST CASE 2

INPUT

30 122 14

OUTPUT

30

```
#include <stdio.h>
int large;
int main() {
      int a,b,c,seclar;
      large =0;
  scanf("%d %d %d",&a,&b,&c);
  if(a>b)
  {
    if(a>c)
      large=a;
    else
      seclar=a;
  }
  if(b>c)
  {   if(b>a)
       large=b;
       seclar=a;
   }
   if(c>a)
     {
       if(c>b)
         large=c;
       else
         seclar=c;
     }
    printf("%d",seclar);
    return(0);
}
```

SESSION: Operators

Q. 26: Computing X

QUESTION DESCRIPTION

The cost price of n articles is the same as the selling price of X articles . If the profit is p% then what is the value of x?

Input format:

The first input is an integer which corresponds to cp (cost price) the second is an integer which corresponds to sp (selling price)

Output Format:

Refer sample Input and output for formatting Specifications

The float values are displayed correct to 2 decimal places.
TEST CASE 1

INPUT
12.5
14.5
OUTPUT
10.92
TEST CASE 2

INPUT
5.0
6.0
OUTPUT
4.72

```
#include<stdio.h>
int main()
{
  float cp,sp,x;
  scanf("%f%f",&cp,&sp);
  x=(cp*100)/(sp+100);
  printf("%.2f",x);

 return 0;
}
```

SESSION: Operators
Q. 27: Plural
QUESTION DESCRIPTION

We need a simple function that determines if a plural is needed or not. It should take a number, and return true if a plural should be used with that number or false if not. This would be useful when printing out a string such as 5 minutes, 14 apples, or 1 sun. All values will be positive integers or floats, or zero
TEST CASE 1

INPUT
1
OUTPUT
false
TEST CASE 2

INPUT
2
OUTPUT
true

```
#include <stdio.h>
int main() {
      int n;
      scanf("%d",&n);
      if (n==1)
      printf("false");
    else
      printf("true");
  return(0);
}
```

SESSION: Operators
Q. 28: LITTLE ELEPHANT AND BALLOONS
QUESTION DESCRIPTION

The Little Elephant from the Zoo of Lviv is going to the Birthday Party of the Big Hippo tomorrow. Now he wants to prepare a gift for the Big Hippo.
He has N balloons, numbered from 1 to N. The i-th balloon has the color Ci and it costs Pi dollars. The gift for the Big Hippo will be any subset (chosen randomly, possibly empty) of the balloons such that the number of different colors in that subset is at least M.
Help Little Elephant to find the expected cost of the gift.

Input
The first line of the input contains a single integer T - the number of test cases. T test cases follow. The first line of each test case contains a pair of integers N and M. The next N lines contain N pairs of integers Ci and Pi, one pair per line.

Output
In T lines print T real numbers - the answers for the corresponding test cases. Your answer will considered correct if it has at most 10^-6 absolute or relative error.

Constraints
1 <= T <= 40
1<= N, Ci <=40
1<= Pi <= 1000000
0 M K, where K is the number of different colors in the test case. "
TEST CASE 1

INPUT

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

OUTPUT

```
11.000000000
7.333333333
```

TEST CASE 2

INPUT

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

```
OUTPUT
12.000000000
3.333333333
```

```
#include<stdio.h>

long long int data[41][2];

long long int Answer(int k,int num);

int main ()
{
    int t,n,m,c,p,i,k;
    long long int answer;
    scanf("%d",&t);
    while(t--)
    {
      scanf("%d%d",&n,&m);
      answer=0;
      for (i=0;i<41;i++)
        data[i][0]=data[i][1]=0;
      for(i=0;i<n;i++)
      {
          scanf("%d%d",&c,&p);
          data[c][0]+=p;
          data[c][1]++;
      }
      for (i=1;i<=40;i++)
        if(data[i]>0)
          answer+=data[i][0]*(1<<(data[i][1]-1))*Answer(m-1,i);
      printf("%.9lf\n",(double)answer/Answer(m,0));
    }
    return 0;
}

long long int Answer(int k, int num)
{
    long long int answer=0,e[41][41],v[41];
    int i,j=1,tot=0;
    for(i=0;i<41;i++)
      if (data[i][0]>0&&i!=num)
      {
        v[j++]=(1<<data[i][1])-1;
        tot++;
      }
    for(i=0;i<tot+1;i++)
      e[i][0]=1;
    for (i=0;i<tot+1;i++)
      for (j=1;j<=tot;j++)
     if (j>i)
       e[i][j]=0;
     else
       e[i][j]=e[i-1][j]+e[i-1][j-1]*v[i];
    for (i=k;i<=tot;i++)
      answer+=e[tot][i];
    return answer;
}
```

SESSION: Operators

Q. 29: Finding bitwise EXOR of two numbers
QUESTION DESCRIPTION

Write a program to find the bitwise EXOR of two decimal numbers.
An EXOR gate reads 2 input either 0 or 1 and outputs 0 if both the inputs are same and outputs 1 if both the inputs are different.
Similarly write a program to read two decimal numbers and finds EXOR of two numbers .
EXAMPLE :
(3) 10 = (011) 2
(5) 10 = (101) 2
EXOR of 3 and 4 is :
(6) 10 = (110) 2
TEST CASE 1

INPUT
12
23
OUTPUT
Bitwise EX-OR of 12 and 23 is=27
TEST CASE 2

INPUT
2
2
OUTPUT
Bitwise EX-OR of 2 and 2 is=0

```
#include <stdio.h>
int main()
{
    int a,b;
    scanf("%d\n%d",&a,&b);
      printf("Bitwise EX-OR of %d and %d is=%d",a,b,a^b);
    return 0;
}
```

SESSION: Operators
Q. 30: Radius of a Circle
QUESTION DESCRIPTION

Madhan is handling mathematics to 8th grade. He taught area and perimeter of geometric shapes to his students. He thought to give a test based on triangle and circles.The task is to calculate radius of the circle that is inscribed in triangle given the three sides of the triangle. He has set 20 questions and he is tired of preparing answer keys.Write a program to find the radius of the circle inscribed in a triangle.

Input and Output Format :
Input consists of three integers a, b and c. The three integer corresponds to three sides of a triangle
TEST CASE 1

INPUT
12 11 7
OUTPUT
Radius=2.53
TEST CASE 2

```
INPUT
7 4 5
OUTPUT
Radius=1.22
```

```
#include <stdio.h>
#include <math.h>
int main() {
      int a,b,c;
      scanf("%d %d %d",&a,&b,&c);
      double radius;
      float s=((a+b+c)*1.0)/2.0;
      radius= sqrt((s-a)*(s-b)*(s-c)/s);
      radius=radius;
      printf("Radius=%.2f",radius);
      return 0;
}
```

SESSION: Control and Looping
Q. 31: Chef and A Large Permutation
QUESTION DESCRIPTION

Today is Chef's birthday. His mom gifted him a truly lovable gift, a permutation of first N positive integers.

She placed the permutation on a very long table in front of Chef and left it for him to play with it. But as there was a lot of people coming and wishing him. It was interfering with his game which made him very angry and he banged the table very hard due to which K numbers from the permutation fell down and went missing.

Seeing her son's gift being spoilt, his mom became very sad. Chef didn't want his mom to be sad as he loves her the most. So to make her happy, he decided to play a game with her with the remaining N - K numbers on the table. Chef wants his mom to win all the games.

Chef and his mom play alternatively and optimally. In Xth move, a player can choose some numbers out of all the numbers available on the table such that chosen numbers sum up to X. After the move, Chosen numbers are placed back on the table.The player who is not able to make a move loses.

Now, Chef has to decide who should move first so that his Mom wins the game.

As Chef is a small child, he needs your help to decide who should move first. Please help him, he has promised to share his birthday cake with you :)

Input
First Line of input contains a single integer T denoting the number of test cases.
First line of each test case contains two space separated integers N and K denoting the size of
permutation and number of numbers fall down from the table.
Next line of each test case contains K space separated integers denoting the values of missing numbers.

Output
For each test case, print "Chef" if chef should move first otherwise print "Mom" (without quotes).

Explanation
For test case 1.

Mom can choose {1} to make 1.
Chef can choose {2} to make 2.
Mom can choose {1,2} to make 3.
Chef can choose {4} to make 4.
Mom can choose {1,4} to make 5.
Chef can choose {2,4} to make 6.
Mom can choose {1,2,4} to make 7.
Chef cannot make 8 out of the numbers on the table.
So,Chef loses and Mom wins.
TEST CASE 1

```
INPUT
2
5 2
3 5
5 1
1
OUTPUT
Mom
Chef
```

TEST CASE 2

```
INPUT
2
4 2
3 4
2 1
3
OUTPUT
Mom
Chef
```

```
#include<stdio.h>

long long x[1000000];

void quicksort(int first,int last){
    int pivot,j,temp,i;

     if(first<last){
         pivot=first;
         i=first;
         j=last;

         while(i<j){
             while(x[i]<=x[pivot]&&i<last)
                 i++;
             while(x[j]>x[pivot])
                 j--;
             if(i<j){
                 temp=x[i];
                  x[i]=x[j];
                  x[j]=temp;
             }
         }
         temp=x[pivot];
         x[pivot]=x[j];
         x[j]=temp;
         quicksort(first,j-1);
         quicksort(j+1,last);

    }
}

int main()
{

    long long t,n,k,i;
    long long rst,sum;
    scanf("%lld",&t);

    while(t--)
    {
        scanf("%lld%lld",&n,&k);
        rst=0;
        sum=0;

        for(i=0;i<k;i++)
        {
            scanf("%lld",&x[i]);
        }


        quicksort(0,k-1);

/*        for(i=0;i<k;i++)
        {
            printf("%lld ",x[i]);
        }
*/


        for(i=0;i<k;i++)
        {
            sum+=x[i];

            if(x[i]>((x[i]*(x[i]+1))/2-sum))
            {
                rst=x[i];
                break;
            }

        }

        if(!rst)
        {
            rst=(n*(n+1))/2-sum+1;
        }


        if(rst%2)
            printf("Chef\n");
        else
            printf("Mom\n");
    }
return 0;
}
```

SESSION: Control and Looping

Q. 32: Devu and friendship testing

QUESTION DESCRIPTION

Devu has n weird friends. Its his birthday today, so they thought that this is the best occasion for testing their friendship with him. They put up conditions before Devu that they will break the friendship unless he gives them a grand party on their chosen day. Formally, ith friend will break his friendship if he does not receive a grand party on dith day.

Devu despite being as rich as Gatsby, is quite frugal and can give at most one grand party daily. Also, he wants to invite only one person in a party. So he just wonders what is the maximum number of friendships he can save. Please help Devu in this tough task !!

Input

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

First line will contain a single integer denoting n.

Second line will contain n space separated integers where ith integer corresponds to the day dith as given in the problem.

Output

Print a single line corresponding to the answer of the problem.

Constraints

1<= T <= 104

1 <= n<= 50

1 <= di<= 100

Explanation

Example case 1. Devu can give party to second friend on day 2 and first friend on day 3, so he can save both his friendships.

Example case 2. Both the friends want a party on day 1, and as the Devu can not afford more than one party a day, so he can save only one of the friendships, so answer is 1.

TEST CASE 1

```
INPUT
2
2
3 2
2
1 1
OUTPUT
2
1
```

TEST CASE 2

```
INPUT
2
4
```

```
3 2 1 4
2
6 9
OUTPUT
4
2
```

```
#include <stdio.h>

int main(void) {
      int i,t,j,n;
      scanf("%d\n",&t);
      while(t--)
      {
          int d[50];
          scanf("%d\n",&n);
          for(i=1;i<=n;i++)
          scanf("%d",&d[i]);
          for(i=1;i<=n;i++)
          {
              for(j=1;j<=n;j++)
              {   if((d[i]==d[j])&&(i!=j))
                  d[j]=0;
              }
          }
          int c=0;
          for(i=1;i<=n;i++)
          {
              if(d[i]!=0)
              c++;
          }
          printf("%d\n",c);
      }
      return 0;
}
```

SESSION: Control and Looping

Q. 33: Holiday Of Equality

QUESTION DESCRIPTION

"In Berland it is the holiday of equality. In honor of the holiday the king decided to equalize the welfare of all citizens in Berland by the expense of the state treasury.

Totally in Berland there are n citizens, the welfare of each of them is estimated as the integer in aiburles (burle is the currency in Berland).

You are the royal treasurer, which needs to count the minimum charges of the kingdom on the king's present. The king can only give money, he hasn't a power to take away them.

Input

The first line contains the integer n (1<n<100) the number of citizens in the kingdom.

The second line contains n integers a1,a2,...,an, where ai (0,ai<106) the welfare of the i-th citizen.

Output

In the only line print the integer S the minimum number of burles which are had to spend.

"

TEST CASE 1

```
INPUT
5
0 1 2 3 4
OUTPUT
10
```

```c
#include<stdio.h>
int main()
{
int ara[100], n, i, max, sum=0;

scanf("%d",&n);

for(i=0;i<n;i++)
{
    scanf("%d", &ara[i]);
}
max=ara[0];
for(i=0;i<n;i++)
{
    if(ara[i]>max)
        max=ara[i];

}
for(i=0;i<n;i++)
{
    sum+=max-ara[i];
}


printf("%d", sum);

return 0;
}
```

SESSION: Control and Looping
Q. 34: Sum of the first and last digit
QUESTION DESCRIPTION

If Give an integer N . Write a program to obtain the sum of the first and last digit of this number

TEST CASE 1

```
INPUT
2
1234
124894
OUTPUT
5
5
```

TEST CASE 2

```
INPUT
1
23233
OUTPUT
5
```

```
#include <stdio.h>

int main(void) {
      int t;
      scanf("%d", &t);
      while(t--)
      {
          int n;
          scanf("%d", &n);
          int k=n;
          int l=n%10;
          int f;
          for(;k>0;k=k/10)
          {
              f=k%10;
          }
          int s=f+l;
          printf("%d\n",s);
      }
      return 0;
}
```

SESSION: Control and Looping
Q. 35: Print 5 Sum of Positive Numbers
QUESTION DESCRIPTION

Write a C program to find allow the user to enter n number and finds the number of positive numbers entered and the sum of all positive numbers entered using a while loop

Input format:

Input consists of n+1 integers. The first integer corresponds to n and the next n integers correspond to the numbers to be added.Consider 0 to be a positive number

Output format:

Refer sample input and output for formatting specifications.

TEST CASE 1

```
INPUT
4
5 -2 -1 6
OUTPUT
Positive Numbers=2
Sum=11
```

TEST CASE 2

```
INPUT
5
-1 -2 -3 -4 -5
OUTPUT
Positive Numbers=0
Sum=0
```

```
#include<stdio.h>
```

```
int main(){
  int n,sum=0,count=0,i=1,b;
  scanf("%d",&n);
  while(i<=n)
  {
    scanf("%d",&b);
    i++;
    if(b>=0)
    {
      count++;
      sum=sum+b;
    }
  }
  printf("Positive Numbers=%d\nSum=%d",count,sum);
  return 0;
}
```

SESSION: Control and Looping
Q. 36: Star formation
QUESTION DESCRIPTION

program to print the half pyramid using star
TEST CASE 1

INPUT
5
OUTPUT

```
*
**
***
****
*****
```

TEST CASE 2

INPUT
3
OUTPUT

```
*
**
***
```

```
#include <stdio.h>
int main() {
      int i,j,n;
      scanf("%d",&n);
      for (i=0;i<=n;i++)
    { printf("\n");
      for (j=1;j<=i;j++){
          printf("*");
        }
    }

return 0;
}
```

SESSION: Control and Looping
Q. 37: Pattern 3
QUESTION DESCRIPTION

Write a program to generate a following @s triangle:

```
@ @ @ @ @
@ @ @ @
@ @ @
@ @
@
```

TEST CASE 1

INPUT

```
7
```

OUTPUT

```
@@@@@@@
@@@@@@
@@@@@
@@@@
@@@
@@
@
```

TEST CASE 2

INPUT

```
5
```

OUTPUT

```
@@@@@
@@@@
@@@
@@
@
```

```
#include <stdio.h>
int main() {
      int i,j,n;
      scanf("%d",&n);
      for(i=n; i>=1; --i)
     {
       printf("\n");
         for(j=1; j<=i; ++j)
         {
           printf("@");
         }
     }

return 0;
}
```

SESSION: Control and Looping

Q. 38: Choose k objects from n distinct objects

QUESTION DESCRIPTION

Description

Choose k objects from n distinct objects

Write a C program that calculates the number of ways to choose k objects from n distinct objects. k and n both are integers.

Input Format:

First line contains the value of n, where 0<=n<=10

Second line contains the value of k, where k>=0

Output Format:

One line containing the number of ways to chose the objects Note: In this question you are not given main() so you have to write the complete program.
TEST CASE 1

INPUT
4
2
OUTPUT
6
TEST CASE 2

INPUT
2
1
OUTPUT
2

```
#include <stdio.h>
long fact(long a);
int main()
{
  int n,k,temp;
  scanf("%d",&n);
  scanf("%d",&k);
  long int d,den1,den2,ans1;
  d=fact(n);
  den1=fact(k);
  temp=n-k;den2=fact(temp);
  ans1=den1*den2;
  ans1=d/ans1;
  printf("%ld",ans1);
  return 0;
}
long fact(long a)
{
  int i;long int ans=1;
  for(i=1;i<=a;i++)
    ans=ans*i;
  return ans;
}
```

SESSION: Control and Looping
Q. 39: PRINT Numbers Within the Range
QUESTION DESCRIPTION

Write a C program to print all numbers between a and b (a and b inclusive) using a while loop.

Input format:

Input consists of 2 integers. The first integer corresponds to a and the second integer corresponds to b . Assume a>=b.

Output format:

Refer sample input and output for formatting specifications.

TEST CASE 1

```
INPUT
8 1
OUTPUT
8 7 6 5 4 3 2 1
```

TEST CASE 2

```
INPUT
9 9
OUTPUT
No Number
```

```
#include <stdio.h>
int main()
{
  int a,b;
  scanf("%d%d",&a,&b);
  if(a==b)
  { printf("No Number");
      return 0;
  }
  while(b<=a)
  {
    printf("%d ",a);
    a--;
  }
  return 0;
}
```

SESSION: Control and Looping
Q. 40: Pattern
QUESTION DESCRIPTION

Print the following pattern :

If N = 1

```
1
```

If N = 2

```
2 2 2
2 1 2
2 2 2
```

If N = 3

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

```
3 2 2 2 3
3 3 3 3 3

and so on.
TEST CASE 1

INPUT
3
OUTPUT
3 3 3 3 3
3 2 2 2 3
3 2 1 2 3
3 2 2 2 3
3 3 3 3 3
TEST CASE 2

INPUT
2
OUTPUT
2 2 2
2 1 2
2 2 2
```

```
#include<stdio.h>
int main()
{

int n,i;
scanf("%d",&n);

int m = 2*n-1;

int arr[50][50];
int j;

for(i=0;i<m;i++){
      for(j=i;j<=m-i-1;j++){
            arr[i][j] = n;
      }
      for(j=i+1;j<=m-i-1;j++){
            arr[j][m-i-1] = n;
      }
      for(j=m-2-i;j>=i;j--){
            arr[m-i-1][j] = n;
      }
      for(j=m-2-i;j>i;j--){
            arr[j][i] = n;
      }
      n--;
}
for(i=0;i<m;i++){
      for(j=0;j<m;j++){
            printf("%d ",arr[i][j]);
      }
      printf("\n");
}
return 0;
}
```

SESSION: Arrays
Q. 41: Array Mean
QUESTION DESCRIPTION

Write a program to find the mean of the elements in the array.

Input and Output Format:

Input consists of n+1 integers where n corresponds to the number of elements in the array.

The first integer corresponds to n and the next n integers correspond to the elements in the array.

Output consists of a double value which corresponds to the mean of the array. It is printed upto 2 digits of precision.

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

Refer sample input and output for formatting specifications.

All text in bold corresponds to input and the rest corresponds to output.
TEST CASE 1

```
INPUT
5
2
4
1
3
5
OUTPUT
The mean of the array is 3.00
```

TEST CASE 2

```
INPUT
10
100
105
200
205
108
15
18
88
1000
12
OUTPUT
The mean of the array is 185.10
```

```
 #include<stdio.h>
int main(){
  int n, a[20], i;
  float min=0;
  scanf("%d",&n);
  for(i=0;i<n;i++)
  {
    scanf("%d",&a[i]);
    min+=a[i];
  }

  printf("The mean of the array is %.2f",min/n);
  return 0;
}
```

SESSION: Arrays
Q. 42: square sum
QUESTION DESCRIPTION

"Everyone knows what a square looks like. Mathematically, a square is a regular quadrilateral. This means that it has four equal sides and four equal angles (90 degree angles).

One beautiful day, Johnny eagerly examined the interesting properties of squares. He did not forget you, his best friend and a talented programmer and thus made a problem about squares to challenge your programming ability. The problem is: given a set of N points in the plane, how many squares are there such that all their corners belong to this set?

Now let's show Johnny your skill!
Input

The first line contains t, the number of test cases (about 10). Then t test cases follow.

Each test case has the following form:

The first line contains an integer N, the number of points in the given set (4 <= N <= 500).
Then N lines follow, each line contains two integers X, Y describing coordinates of a point (-50,= X, Y <= 50).

Output

For each test case, print in a single line the number of squares that have vertices belong to the given set."
TEST CASE 1

```
INPUT
1
7
0 0
0 1
1 0
1 1
1 2
2 1
2 2
OUTPUT
3
```

TEST CASE 2

```
INPUT
1
4
1 1
1 2
2 1
```

```
2 2
OUTPUT
1
```

```
 #include <stdio.h>
        #define  z 
(d3>=0)&&(d3<=100)&&(a3>=0)&&(a3<=100)&&(point[d3][a3]==1)&&(d4>=0)&&(d4<
=100)&&(a4>=0)&&(a4<=100)&&(point[d4][a4]==1)

        int main()  {

            int numcase,I,i,j,n,count,x[500],y[500],point[101][101];
            int d1,d2,d3,d4,a1,a2,a3,a4;

            scanf("%d",&numcase);
            for(I=0;I<numcase;I++)  {
                scanf("%d",&n);
                for(i=0;i<=100;i++)
                    for(j=0;j<=100;j++)  point[i][j]=0;
                for(i=0;i<n;i++)  {
                    scanf("%d%d",&x[i],&y[i]);
                    x[i]+=50;
                    y[i]+=50;
                    point[x[i]][y[i]]=1;
                }
                count=0;
                for(i=0;i<n;i++)  {
                    for(j=0;j<n;j++)  {
                        if(i!=j)  {

                           d1=x[i],a1=y[i];
                           d2=x[j],a2=y[j];

                           d3=d1+a1-a2;
                           a3=a1+d2-d1;

                           d4=d2+a1-a2;
                           a4=a2+d2-d1;


        if(z)
           count++;
             }}}
                printf("%d\n",count/4);
           }
           return 0;
        }
```

SESSION: Arrays

Q. 43: LITTLE ELEPHANT AND BALLOONS

QUESTION DESCRIPTION

"The Little Elephant from the Zoo of Lviv is going to the Birthday Party of the Big Hippo tomorrow. Now he wants to prepare a gift for the Big Hippo.

He has N balloons, numbered from 1 to N. The i-th balloon has the color Ci and it costs Pi dollars. The gift for the Big Hippo will be any subset (chosen randomly, possibly empty) of the balloons such that the number of different colors in that subset is at least M.

Help Little Elephant to find the expected cost of the gift.

Input

The first line of the input contains a single integer T - the number of test cases. T test cases follow. The first line of each test case contains a pair of integers N and M. The next N lines contain N pairs of integers Ci and Pi, one pair per line.

Output

In T lines print T real numbers - the answers for the corresponding test cases. Your answer will considered correct if it has at most 10^-6 absolute or relative error.

Constraints

1 <= T <= 40
1 <= N, Ci <=40
1<= Pi <= 1000000
0 M K, where K is the number of different colors in the test case.
"
TEST CASE 1

```
INPUT
2
2 2
1 4
2 7
2 1
1 4
2 7
OUTPUT
11.000000000
7.333333333
```

TEST CASE 2

```
INPUT
2
2 2
5 5
2 7
2 1
1 4
2 1
OUTPUT
12.000000000
3.333333333
```

```c
#include<stdio.h>

int main()
{
      int t;
      scanf("%d",&t);

      int n,k;

      long long int potwo[43];
      potwo[0]=1;

      int i,j;
      for(i=1;i<43;i++)
      {
            potwo[i]=potwo[i-1]*2;
      }

      long long int sum[41];

      int colour;

      long long int price;

      int number[41];

      struct knap {long long int sumtill;long long int perm;};

      struct knap ar[41][41];

      long long int tempperm;

      while(t--)
      {

            scanf("%d %d",&n,&k);

            for(i=0;i<=40;i++){sum[i]=0;number[i]=0;}

            for(i=0;i<n;i++)
            {
                  scanf("%d %lld",&colour,&price);

                  sum[colour]=2*sum[colour]+potwo[number[colour]]*price;

                  number[colour]++;
            }

            for(i=1;i<=40;i++)
            {
                  for(j=1;j<=40;j++)
                  {
                        ar[i][j].sumtill=0;ar[i][j].perm=0;
                  }
            }
/*
            for(i=1;i<=40;i++)

            {
                  printf("FOR %d:%d %lld\n",i,number[i],sum[i]);
            }
*/
            ar[1][0].sumtill=0;
            ar[1][0].perm=1;
            ar[1][1].sumtill=sum[1];
            ar[1][1].perm=potwo[number[1]]-1;

//          printf("%lld %lld %lld
%lld\n",ar[1][0].sumtill,ar[1][0].perm,ar[1][1].sumtill,ar[1][1].perm);

            for(i=2;i<=40;i++)
            {
                  tempperm=potwo[number[i]]-1;
                  ar[i][0].sumtill=0;
                  ar[i][0].perm=1;
                  for(j=1;j<i;j++)
```

```
                 {
                       ar[i][j].sumtill=ar[i-1][j].sumtill+ar[i-1][j-
1].sumtill*tempperm+sum[i]*ar[i-1][j-1].perm;
                       ar[i][j].perm=ar[i-1][j].perm+tempperm*ar[i-1][j-
1].perm;
                 }

                                 ar[i][j].sumtill=ar[i-1][j-
1].sumtill*tempperm+sum[i]*ar[i-1][j-1].perm;
                       ar[i][j].perm=tempperm*ar[i-1][j-1].perm;

           }
/*
           for(i=1;i<=2;i++)
           {
                 for(j=0;j<=i;j++)
                 {
                       printf("%d %d:%lld %lld
\n\n",i,j,ar[i][j].sumtill,ar[i][j].perm);
                 }
           }
*/
           long long int nume=0,deno=0;

           for(j=k;j<=40;j++)
           {
                 nume+=ar[40][j].sumtill;
                 deno+=ar[40][j].perm;
           }

           printf("%.9lf\n",(1.0*nume)/deno);

      }

      return 0;
}
```

SESSION: Arrays
Q. 44: uncle jony
QUESTION DESCRIPTION

Vlad enjoys listening to music. He lives in Sam's Town. A few days ago he had a birthday, so his parents gave him a gift: MP3-player! Vlad was the happiest man in the world! Now he can listen his favorite songs whenever he wants!
Vlad built up his own playlist. The playlist consists of N songs, each has a unique positive integer length. Vlad likes all the songs from his playlist, but there is a song, which he likes more than the others. It's named ""Uncle Johny"".
After creation of the playlist, Vlad decided to sort the songs in increasing order of their lengths. For example, if the lengths of the songs in playlist was {1, 3, 5, 2, 4} after sorting it becomes {1, 2, 3, 4, 5}. Before the sorting, ""Uncle Johny"" was on K-th position (1-indexing is assumed for the playlist) in the playlist.
Vlad needs your help! He gives you all the information of his playlist. Your task is to find the position of ""Uncle Johny"" in the sorted playlist.
Input

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 one integer N denoting the number of songs in Vlad's playlist. The second line contains N space-separated integers A1, A2, ..., AN denoting the lenghts of Vlad's songs.
The third line contains the only integer K - the position of ""Uncle Johny"" in the initial playlist.

Output
For each test case, output a single line containing the position of ""Uncle Johny"" in the sorted playlist.

Constraints
1 <=T <= 1000
1 <= K N <= 100
1 <= Ai <= 109

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

inline int fast_scan()
{
    int n=0;
    int ch=getchar_unlocked();
    while( ch <48 )ch=getchar_unlocked();
    while( ch >47 )
        n = (n<<3)+(n<<1) + ch-'0', ch=getchar_unlocked();
    return n;
}

void quickSort(int arr[], int left, int right) {
      int i = left, j = right;
      int tmp;
      int pivot = arr[(left + right) / 2];

      /* partition */
      while (i <= j) {
            while (arr[i] < pivot)
                  i++;

            while (arr[j] > pivot)
                  j--;

            if (i <= j) {
                  tmp = arr[i];
                  arr[i] = arr[j];
                  arr[j] = tmp;
                  i++;
                  j--;
            }
      };

      /* recursion */

      if (left < j)
            quickSort(arr, left, j);

      if (i < right)
```

```
           quickSort(arr, i, right);
}

int main()
{
    int numCases = fast_scan();

    while(numCases--)
    {
        int numSongs = fast_scan();
        int lengths[numSongs];
        int i = 0;
        int j,k,val;
         j = numSongs;

        while(j--)
        {
            lengths[i++] = fast_scan();
        }

        k = fast_scan();
        val = lengths[k-1];

        quickSort(lengths, 0, numSongs-1);

        for(i = 0; i < numSongs; i++)
        {
            if(lengths[i] == val)
            {
                printf("%d\n",i+1);
                break;
            }
        }
    }

    return 0;
}
```

SESSION: Arrays
Q. 45: Array Transformation
QUESTION DESCRIPTION

"Given n numbers, you can perform the following operation any number of times : Choose any subset of the numbers (possibly empty), none of which are 0. Decrement the numbers in the subset by 1, and increment the numbers not in the subset by K.

Is it possible to perform operations such that exactly n - 1 numbers become 0 ?
TEST CASE 1

```
INPUT
3
2 1
10 10
3 2
1 2 2
3 2
1 2 3
OUTPUT
```

```
YES
YES
NO
```

TEST CASE 2

INPUT

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

OUTPUT

```
YES
YES
YES
```

```c
#include <stdio.h>
int main()
{
  int t,n,k,a[100],hash[1000],i,ans,x;
  scanf("%d",&t);
  while(t--)
  {
    scanf("%d %d",&n,&k);
    for(i=0;i<1000;i++)hash[i]=0;
    ans=0;
    for(i=0;i<n;i++)
    {
      scanf("%d",&a[i]);
      x=(a[i]%(k+1));
      hash[x]++;
      if(hash[x]>n-2)ans=1;
    }
    if(ans)
      printf("YES\n");
    else
      printf("NO\n");
  }
  return 0;
}
```

SESSION: Arrays

Q. 46: Arrsumavg

QUESTION DESCRIPTION

Calculate Sum & Average of an Array

Assign the maximum size value is 10

TEST CASE 1

INPUT

```
6
-8 9 -100 0 6 5
```

OUTPUT

```
Sum=-108
Sum=20
Average=-14.67
```

```
TEST CASE 2

INPUT
4
-8 9 -100 0
OUTPUT
Sum=-108
Sum=9
Average=-24.75
```

```
#include <stdio.h>
#define MAXSIZE 10

int main()
{

        int array[MAXSIZE];
      int i, num, negative_sum = 0, positive_sum = 0;
      float total = 0.0, average;


      scanf("%d", &num);


        for (i = 0; i < num; i++)
        {
           scanf("%d", &array[i]);
      }


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

           if (array[i] < 0)
                {
                 negative_sum = negative_sum + array[i];
           }
           else if (array[i] > 0)
                {
                 positive_sum = positive_sum + array[i];
           }
           else if (array[i] == 0)
                {
                 ;
           }
           total = total + array[i] ;

        }

        average = total / num;

        printf("\nSum=%d\n", negative_sum);

        printf("Sum=%d\n", positive_sum);

        printf("Average=%.2f\n", average);

      return 0;
}
```

SESSION: Arrays
Q. 47: game
QUESTION DESCRIPTION

" You are playing following game: given an array A of N natural numbers. All numbers in the array A are at most M. On every turn you may pick any two different elements Ai and Aj (ij), such that Ai, Aj M, and add K to both. The game ends when you are not able to continue. That is, when there is no pair (i,j) left such that both of them are less than equal to M.

Let's call two arrays different if the sum of all their elements is different. When the game ends, you note down the final array A. How many different final arrays can you have.
Input

The first line contains three integers N, M and K. N elements of the array follow in the next line.
Constraints

1 <= N <= 1000000
1 <= M,K<= 1000000000000
1 <= Ai <= M
test case 1:All possible sums are 14 and 10. You can get them by, for example, these arrays:

A=(5, 4, 5),

A=(1, 4, 5)

The above arrays are different because their sums are different."
TEST CASE 1

INPUT
3 3 2
1 2 3
OUTPUT
2
TEST CASE 2

INPUT
3 3 3
2 2 2
OUTPUT
1

```
#include<stdio.h>
int N;
long long M,K;
long long A[100];
int main()
{
    int i;
    long long sum=0;
    long long max=0;
    scanf("%d %lld %lld",&N,&M,&K);
    for(i=0;i<N;i++)
    {
```

```
    scanf("%lld",&A[i]);
    A[i]=(M-A[i])/K+1;

    if(max<A[i])
    max=A[i];

    sum+=A[i];
    }
    long long min=0;

    if((sum-max)%2==0)
    min=(sum-max)/2;
    else
    min=(sum-max)/2+1;

    long long ans=sum/2-min+1;
    ans=ans%(1000000007);
    printf("%lld\n",ans);

    return 0;
}
```

SESSION: Arrays
Q. 48: SYMMETRIC MATRIX
QUESTION DESCRIPTION

A symmetric matrix is a square matrix that is equal to its transpose.

Write a C program to find whether a given matrix is a square matrix or not.
Input Format:

The input consists of (m*n+2) integers. The first integer corresponds to m, the number of rows in the matrix and the second integer corresponds to n, the number of columns in the matrix. The remaining integers correspond to the elements in the matrix. The elements are read in rowwise order, first row first, then second row and so on. Assume that the maximum value of m and n is 10
TEST CASE 1

```
INPUT
3
3
1 2 3
4 5 6
7 8 9
OUTPUT
Not Symmetric
```

TEST CASE 2

```
INPUT
3
3
3 1 1
1 3 1
1 1 5
OUTPUT
Symmetric
```

```
#include<stdio.h>
```

```
int main()
{
  int m, n, c, d, matrix[10][10], transpose[10][10];
  scanf("%d\n%d", &m, &n);

  for (c = 0; c < m; c++)
    for (d = 0; d < n; d++)
      scanf("%d", &matrix[c][d]);

  for (c = 0; c < m; c++)
    for (d = 0; d < n; d++)
      transpose[d][c] = matrix[c][d];

  if (m == n)
  {
    for (c = 0; c < m; c++)
    {
      for (d = 0; d < m; d++)
      {
        if (matrix[c][d] != transpose[c][d])
          break;
      }
      if (d != m)
        break;
    }
    if (c == m)
      printf("Symmetric\n");
    else
      printf("Not Symmetric\n");
  }
  else
    printf("Symmetric\n");

  return 0;
}
```

QUESTION
SESSION: Arrays
Q. 49: Magic Square
QUESTION DESCRIPTION

A magic square is an arrangement of numbers (usually integers) in a square grid, where the numbers in each row, and in each column, and the numbers in the forward and backward main diagonals, all add up to the same number

Input Format:

The input consists of (n*n+1) integers. The first integer corresponds to the number of rows/columns in the matrix. The remaining integers correspond to the elements in the matrix. The elements are read in rowwise order, first row first, then second row and so on. Assume that the maximum value of m and n is 5.

Output Format:
Print yes if it is a magic square. Print no if it is not a magic square
TEST CASE 1

```
INPUT
4 9 2
3 5 7
8 1 6
OUTPUT
Yes
TEST CASE 2

INPUT
4 9 2
3 5 7
8 1 1
OUTPUT
No
```

```c
#include<stdio.h>

int main() {
   int size = 3;
   int matrix[3][3];
   int row, column = 0;
   int sum, sum1, sum2;
   int flag = 0;

   for (row = 0; row < size; row++) {
      for (column = 0; column < size; column++)
         scanf("%d", &matrix[row][column]);
   }

   sum = 0;
   for (row = 0; row < size; row++) {
      for (column = 0; column < size; column++) {
         if (row == column)
            sum = sum + matrix[row][column];
      }
   }

   for (row = 0; row < size; row++) {
      sum1 = 0;
      for (column = 0; column < size; column++) {
         sum1 = sum1 + matrix[row][column];
      }
      if (sum == sum1)
         flag = 1;
      else {
         flag = 0;
         break;
      }
   }

   for (row = 0; row < size; row++) {
      sum2 = 0;
      for (column = 0; column < size; column++) {
         sum2 = sum2 + matrix[column][row];
      }
      if (sum == sum2)
         flag = 1;
      else {
         flag = 0;
         break;
      }
   }

   if (flag == 1)
      printf("Yes");
   else
      printf("No");

   return 0;
}
```

QUESTION
SESSION: Arrays
Q. 50: Subsequence
QUESTION DESCRIPTION

"Finding the longest increasing subsequence is an old and well-known problem now. Here you will have to do something similar. You need to find the longest weird subsequence (LWS) of the given string. The subsequence is called weird if it can be split into two disjoint subsequences, one of which is non-decreasing and the other one is non-increasing.
Just for clarity, by subsequence of the given string S we mean any string that can be obtained from S by erasing from it zero or more characters. So empty string is a subsequence of any string and any string is a subsequence of itself. Further, note that we consider only strings composed of lowercase Latin letters and these letters compared by their ASCII codes. So, for example, 'a' is smaller than 'b' and 'p' is larger than 'h'.
Now let's consider some example. Let S=""aabcazcczba"". Then ""abczz"" is its some non-decreasing subsequene, ""zccb"" is its some non-increasing subsequence and ""aabczcczba"" is its some weird subsequence since it can be split into non-decreasing subsequence ""aabzz"" and non-increasing subsequence ""cccba"": ""AABcZccZba"" (first subsequence is shown by capital letters just for calrity).
Input
The first line contains a single positive integer T, the number of test cases. T test cases follow. The only line of each test case contains a non-empty string S composed of lowercase Latin letters.
Output
For every test case, output the length of the LWS of the given string.
"
TEST CASE 1

```
INPUT
3
abc
cbazyzabc
ddaabbaacc
OUTPUT
3
6
10
```

TEST CASE 2

```
INPUT
1
hadsadchgs
```

```
OUTPUT
8
```

```
#include<stdio.h>
#include<string.h>
int main()
{
 int t;
 scanf("%d",&t);
 while(t--)
 {
  char s[2005];
  int a[26][26];
  int i,j;
   for(i=0;i<26;i++)
     for(j=0;j<26;j++)
       a[i][j]=0;
  scanf("%s",s);
  int k,max;
  int len=strlen(s);
  int first[26]={0};
  int second[26]={0};
  for(i=0;i<len;i++)
  {
  int x=s[i]-'a';
  for(k=0;k<26;k++)
  {
   max=-1;
   for(j=0;j<=x;j++)
  if(a[j][k]>max)
   max=a[j][k];
   first[k]=max+1;
  }
  for(j=0;j<26;j++)
  {
  max=-1;
  for(k=25;k>=x;k--)
   if(a[j][k]>max)
    max=a[j][k];
  second[j]=max+1;
  }
  for(j=0;j<26;j++)
   a[x][j]=first[j];
  for(j=0;j<26;j++)
   a[j][x]=second[j];
 }

  int ans=-1;
  for(i=0;i<26;i++)
 for(j=0;j<26;j++)
  if(a[i][j]>ans)
   ans=a[i][j];

 printf("%d\n",ans);
}
return 0;
}
```

SESSION: String
Q. 51: Symbols Filter
QUESTION DESCRIPTION

Ganga found a diary, she cant understand what is written in it. Because the letters are mingled with special symbols. She needs to filter those letters to read that diary. can you help her?
TEST CASE 1

INPUT
pass@word
OUTPUT
password
TEST CASE 2

INPUT
wel$co*me
OUTPUT
welcome

```
#include<stdio.h>
int main()
{
    char line[150];
    int i, j;
    for(i=0;i<10;i++)
           scanf("%s",&line[i]);

    for(i = 0; line[i] != '\0'; ++i)
    {
        while (!( (line[i] >= 'a' && line[i] <= 'z') || (line[i] >= 'A'
&& line[i] <= 'Z') || line[i] == '\0') )
        {
            for(j = i; line[j] != '\0'; ++j)
            {
                line[j] = line[j+1];
            }
            line[j] = '\0';
        }
    }
    puts(line);
    return 0;
}
```

QUESTION
SESSION: String
Q. 52: Deleting Vowels
QUESTION DESCRIPTION

Vowels includes only five letter that give great trouble to us. They are mingled with other letters in a string. Shanthi has a task to filter vowels from that string. Help her to do this task.
TEST CASE 1

INPUT
hai
OUTPUT
h
TEST CASE 2

INPUT
man
OUTPUT
mn

```
#include <stdio.h>
#include <string.h>
#define MAX_LIMIT 20

int check_vowel(char);

int main()
{
  char s[100], t[100];
  int c, d = 0;
  fgets(s, MAX_LIMIT, stdin);

  for(c = 0; s[c] != '\0'; c++)
  {
    if(check_vowel(s[c]) == 0)
    {
      t[d] = s[c];
      d++;
    }
  }

  t[d] = '\0';

  strcpy(s, t);

  printf("%s", s);

  return 0;
}

int check_vowel(char ch)
{
    if (ch == 'a' || ch == 'A' || ch == 'e' || ch == 'E' || ch == 'i' ||
ch == 'I' || ch =='o' || ch=='O' || ch == 'u' || ch == 'U')
      return 1;
    else
      return 0;
}
```

QUESTION
SESSION: String
Q. 53: Counting Character
QUESTION DESCRIPTION

Ram went to temple and his uncle asked him to count the number of characters in a text.Write a C code to help rohan count the number of characters in a text.
TEST CASE 1

INPUT
hai
OUTPUT
3
TEST CASE 2

INPUT
welcomeking
OUTPUT
11

```
#include <stdio.h>
int main()
{
    char s[15];
    int i;

    scanf("%s", s);

    for(i = 0; s[i] != '\0'; ++i);

    printf("%d", i);
    return 0;
}
```

SESSION: String
Q. 54: ENCRYPTION
QUESTION DESCRIPTION

Cryptography is a technique that helps to implement security. Flora loves cryptography , She sends a string that replaces each characters with its third letter. Help her to write a code to read the string. Add each characters ascii value of the string with the position of the character present in the string. Print the above manipulated value as well as the manipulated character.
TEST CASE 1

INPUT
hello
OUTPUT
igopt
TEST CASE 2

INPUT
hai
OUTPUT
icl

```
#include <stdio.h>
#include<string.h>
int main()
{
  char c[20];
  int x,i;
  scanf("%s",c);
  x=strlen(c);
  for(i=0;i<x;i++)
  {
    c[i]=c[i]+(i+1);
  }
  printf("%s",c);
 return 0;
}
```

QUESTION
SESSION: String
Q. 55: Wordstem
QUESTION DESCRIPTION

Given n words w[1..n], which originate from the same stem (e.g. grace, graceful, disgraceful, gracefully), we are interested in the original stem. To simplify the problem, we define the stem as the longest consecutive substring that occurs in all the n words. If there are ties, we will choose the smallest one in the alphabetical (lexicographic) order.

TEST CASE 1

INPUT
```
1
4
grace graceful disgraceful gracefully
```
OUTPUT
```
grace
```
TEST CASE 2

INPUT
```
1
2
courageful courage
```
OUTPUT
```
courage
```

```c
#include <stdio.h>
#include<string.h>
int main()
{
  int i,j,k,l,m,n,o,p,t,len[15],flag,s,z,key,min;
  char a[15][25],ans[25],aa[25];
  scanf("%d",&t);
  for(z=1;z<=t;z++)
  {
    flag=s=0;
    min=1000;
    for(i=0;i<=24;i++)
    {
      ans[i]='\0';
    }
    scanf("%d",&n);
    for(i=0;i<=n-1;i++)
    {
      scanf("%s",a[i]);
      m=strlen(a[i]);
      len[i]=m;
      if(m<min)
      {
        min=m;
        key=i;
      }
    }
    for(i=min;i>=1;i--)
    {
      for(j=0;j<=min-i;j++)
      {
        l=j;
        k=0;
        while(k<=i-1)
        {
          aa[k]=a[key][l];
          k++;
          l++;
        }
        aa[k]='\0';
        for(o=0;o<=n-1;o++)
        {
          flag=0;
          for(p=0;p<=len[o]-i;p++)
          {
            if(strcmp(&a[o][p],aa)==a[o][p+i])
            {
              flag=1;
              break;
            }
          }
          if(flag==0)
            break;
        }
        if(flag==1)
        {
          s=1;
          if(strlen(ans)==0)
          {
            strcpy(ans,aa);
          }
          else
          {
            if(strcmp(ans,aa)>0)
              strcpy(ans,aa);
          }
        }
      }
      if(s==1)
        break;
    }
    printf("%s\n",ans);
  }
  return 0;
}
```

SESSION: String
Q. 56: Reverse string
QUESTION DESCRIPTION

"If an Integer N, write a program to reverse the given number.
Input

The first line contains an integer T, total number of testcases. Then follow T lines, each line contains an integer N.
Output

Display the reverse of the given number N.
Constraints

1 <= T<= 1000
1 <= N <= 100000
"

TEST CASE 1

```
INPUT
3
12345
31203
2123
OUTPUT
54321
30213
3212
```

TEST CASE 2

```
INPUT
2
78657
100034
OUTPUT
75687
430001
```

```
#include<stdio.h>
int main()
{
int t,n,r;
scanf("%d\n",&t);
while(t--)
{
scanf("%d\n",&n);
r=0;
while(n)
      {
            r=r*10+ n%10 ;
            n=n/10;

      }
      printf("%d\n",r);
      }
      return 0;
}
```

SESSION: String
Q. 57: String Pattern
QUESTION DESCRIPTION

Given a string
char str[ ] =123456789;
Write a program that displays the following:

```
1
2 3 2
3 4 5 4 3
4 5 6 7 6 5 4
5 6 7 8 9 8 7 6 5
```

TEST CASE 1

```
INPUT
5
OUTPUT
1
2 3 2
3 4 5 4 3
4 5 6 7 6 5 4
5 6 7 8 9 8 7 6 5
TEST CASE 2

INPUT
3
OUTPUT
1
2 3 2
3 4 5 4 3
```

```c
#include <stdio.h>

int main()
{
      int n, c, d, num = 1, space;

      scanf("%d",&n);

      space = n - 1;

      for ( d = 1 ; d <= n ; d++ )
      {
          num = d;


          for ( c = 1 ; c <= d ; c++ )
          {
              printf("%d ", num);
              num++;
          }
          num--;
          num--;
          for ( c = 1 ; c < d ; c++)
          {
              printf("%d ", num);
              num--;
          }
          printf("\n");

      }

      return 0;
}
```

SESSION: String
Q. 58: Reverse the string
QUESTION DESCRIPTION

A and B are friends. They decide to play a game which is one person is writing a group of words or numbers to form a sentence and other person has to reverse each word of that sentence.
TEST CASE 1

INPUT
c programming world
OUTPUT
c gnimmargorp dlrow
TEST CASE 2

INPUT
abcd
OUTPUT
dcba

```
#include <stdio.h>
#include<string.h>
int main()
{
  int i,j=0,k=0,len,x;
  char str[100],str1[10][20],temp;
  scanf("%[^\n]s",str);

  for(i=0;str[i]!='\0';i++)
  {
    if(str[i] == ' ')
    {
      str1[k][j]='\0';
      k++;
      j=0;
    }
    else
    {
      str1[k][j]=str[i];
      j++;
    }
  }
  str1[k][j]='\0';

  for(i=0;i<=k;i++)
  {
    len=strlen(str1[i]);
    for(j=0,x= len - 1; j<x; j++, x--)
    {
      temp=str1[i][j];
      str1[i][j]=str1[i][x];
      str1[i][x]=temp;
    }
  }
  for(i=0;i<=k;i++)
  {
    printf("%s ",str1[i]);
  }
  return 0;
}
```

SESSION: String
Q. 59: Replace a string
QUESTION DESCRIPTION

Write a C program to replace all occurrence of a character with another character using function

The Input is as follows:
1. The string input
2. The character to be replaced in the input string
3. The NEW Character to be replaced in the input string
TEST CASE 1

INPUT
srasrasra
a
m
OUTPUT
srmsrmsrm
TEST CASE 2

INPUT
mmdmm
m
a
OUTPUT
aadaa

```
#include<stdio.h>
#include <string.h>
int main()
{
       char ch[100];
       char a[100],b[100];
       int i,l;
       scanf("%s",ch);
       scanf("%s",a);
       scanf("%s",b);
       //printf("%c%c",a,b);
       l=strlen(ch);
       for(i=0;i<l;i++)
       {
         if(ch[i]==a[0])
           printf("%c",b[0]);
         else
           printf("%c",ch[i]);
       }
       return 0;
     }
```

SESSION: String
Q. 60: Word Index 4
QUESTION DESCRIPTION

Write a C program to count occurrences of a word in a given string using loop.
TEST CASE 1

INPUT
srm university srm university srm university
srm
OUTPUT
srm=3
TEST CASE 2

```
INPUT
care ulc srm
ulc
OUTPUT

ulc=1
```

```c
#include <stdio.h>
#include <string.h>

char str[100], sub[100];
int count = 0, count1 = 0;

int main()
{
    int i, j, l1, l2;

    scanf("%[^\n]s", str);

    l1 = strlen(str);

    scanf(" %[^\n]s", sub);

    l2 = strlen(sub);

    for (i = 0; i < l1;)
    {
        j = 0;
        count = 0;
        while ((str[i] == sub[j]))
        {
            count++;
            i++;
            j++;
        }
        if (count == l2)
        {
            count1++;
            count = 0;
        }
        else
            i++;
    }
    printf("%s=%d", sub, count1);
      return 0;
}
```

SESSION: Functions
Q. 61: Compare 2 arrays using Function
QUESTION DESCRIPTION

Write a program to find whether 2 arrays are the same.

Input Format:

Input consists of 2n+1 integers. The first integer corresponds to n , the size of the array. The next n integers correspond to the elements in the first array. The next n integers correspond to the elements in the second array.Assume that the maximum value of n is 15.

Output Format:

Print yes if the 2 arrays are the same. Print no if the 2 arrays are different.

TEST CASE 1

```
INPUT
5
2 3 6 8 -1
2 3 6 8 -1
OUTPUT
yes
```

TEST CASE 2

```
INPUT
5
2 3 6 8 -1
2 3 6 8 10
OUTPUT
no
```

```c
#include<stdio.h>
int main()
{
  int a[100],n,i,b[100],c=0;
  scanf("%d",&n);
  for(i=0;i<n;i++)
    scanf("%d",&a[i]);
  for(i=0;i<n;i++)
  {
    scanf("%d",&b[i]);
    if(a[i]!=b[i])
      c=1;
  }
  if(c==1)
    printf("no");
  else
    printf("yes");
  return 0;
}
```

SESSION: Functions

Q. 62: Sticks

QUESTION DESCRIPTION

Chef and his little brother are playing with sticks. They have total N sticks. Length of i-th stick is Ai. Chef asks his brother to choose any four sticks and to make a rectangle with those sticks its sides. Chef warns his brother to not to break any of the sticks, he has to use sticks as a whole. Also, he wants that the rectangle formed should have the maximum possible area among all the rectangles that Chefs brother can make.

Chefs little brother takes this challenge up and overcomes it. Can you also do so? That is, you have to tell whether it is even possible to create a rectangle? If yes, then you have to tell the maximum possible area of rectangle.

Input

The first line contains a single integer T denoting the number of test-cases. T test cases follow.

The first line of each test case contains a single integer N denoting the number of sticks.

The second line of each test case contains N space-separated integers A1, A2, ..., AN denoting the lengths of sticks.
Output

For each test case, output a single line containing an integer representing the maximum possible area for rectangle or -1 if its impossible to form any rectangle using the available sticks.
Constraints

1 <= T <= 100
1 <= N <= 103
1 <= sum of Ns over all test-cases in a single test file <= 103
1 <= Ai <= 103
TEST CASE 1

INPUT
2
4
2 3 1 2
4
1 2 2 3
OUTPUT
-1
-1
TEST CASE 2

INPUT
1
7
1 2 3 4 5 6 7
OUTPUT
-1

```
#include<stdio.h>
int main(){
      int t,n,i,j,k;
      int arr[1000],a[2]={0};
      scanf("%d",&t);
      for(i=0;i<t;i++){
            a[0]=a[1]=0;
            scanf("%d",&n);
            for(j=0;j<n;j++){
                  scanf("%d",&arr[j]);
            }
            for(j=0;j<n;j++){
                  if(arr[j]!=0){
                        for(k=j+1;k<n;k++){
                            if(arr[j]==arr[k]){
                              if(arr[j]>a[0]){
                                    a[1]=a[0];
                                    a[0]=arr[j];
                                      arr[j]=0;arr[k]=0;
                                    }
                                    else if(arr[j]>a[1]){
                                         a[1]=arr[j];
                                         arr[j]=0;arr[k]=0;
                                   }
                                   break;
                            }
                      }
                  }
            }
            if((a[0]!=0)&&(a[1]!=0)){
                  printf("%d\n",a[0]*a[1]);
            }
            else{
                  printf("-1\n");
            }
      }
      return 0;
}
```

QUESTION
SESSION: Functions
Q. 63: Caravan
QUESTION DESCRIPTION

Most problems on SRM University eLab highlight eLab love for programming but little is known about his love for racing sports. He is an avid Formula 1 fan.

He went to watch this years Indian Grand Prix at New Delhi. He noticed that one segment of the circuit was a long straight road. It was impossible for a car to overtake other cars on this segment. Therefore, a car had to lower down its speed if there was a slower car in front of it. While watching the race, Chef started to wonder how many cars were moving at their maximum speed.

Formally, youre given the maximum speed of N cars in the order they entered the long straight segment of the circuit. Each car prefers to move at its maximum speed. If thats not possible because of the front car being slow, it might have to lower its speed. It still moves at the fastest possible speed while avoiding any collisions. For the purpose of this problem, you can assume that the straight segment is infinitely long.

Count the number of cars which were moving at their maximum speed on the straight segment.
Input

The first line of the input contains a single integer T denoting the number of test cases to follow. Description of each test case contains 2 lines. The first of these lines contain a single integer N, the number of cars. The second line contains N space separated integers, denoting the maximum speed of the cars in the order they entered the long straight segment.
Output

For each test case, output a single line containing the number of cars which were moving at their maximum speed on the segment.
TEST CASE 1

INPUT

```
3
1
10
3
8 3 6
5
4 5 1 2 3
OUTPUT
1
2
2
TEST CASE 2

INPUT
2
5
7 2 8 3 6
3
88 6 5
OUTPUT
2
3
```

```
#include <stdio.h>

int main(void) {
      int T,i,j,k;
    scanf("%d",&T);
    for(i=0;i<T;i++)
    {
        int N;
        scanf("%d",&N);
        int arr[N];
        for(j=0;j<N;j++)
        {
            int a;
            scanf("%d",&a);
            arr[j]=a;
        }
        int ans=1,f=0;
        for(j=1;j<N;j++)
        {  f=0;
            for(k=0;k<j;k++)
            {
                if (arr[k]<arr[j])
                  f=1;
            }
            if (f!=1)
            ans++;
        }
        printf("%d\n",ans);
    }
      return 0;
}
```

QUESTION
SESSION: Functions
Q. 64: Recursion 8 : Maximum Element in Array
QUESTION DESCRIPTION

Write a program to find the maximum element in an array using recursion.

Input and Output Format:

Input consists of n+1 integers.

Refer sample input and output for formatting specifications.

All text in bold corresponds to input and the rest corresponds to output.

TEST CASE 1

```
INPUT
6
2 5 1 7 4 2
OUTPUT
Maximum element in the array is 7
```

TEST CASE 2

```
INPUT
4
9 8 7 6
OUTPUT
Maximum element in the array is 9
```

```
#include <stdio.h>

int main()
{
  int array[100], maximum, size, c, location = 1;

  scanf("%d", &size);

  for (c = 0; c < size; c++)
    scanf("%d", &array[c]);

  maximum = array[0];

  for (c = 1; c < size; c++)
  {
    if (array[c] > maximum)
    {
       maximum  = array[c];
       location = c+1;
    }
  }

  printf("Maximum element in the array is %d", maximum);
  return 0;
}
```

SESSION: Functions
Q. 65: Powers of a number
QUESTION DESCRIPTION

Pavithra asks a question to Sumathi to find the power of a given number. She explains wht she has to do. She says"Get 2 values from the user a power n and use a function power(a,n)". Can you write a code for this same scenario.

TEST CASE 1

INPUT
10 3
OUTPUT
1000.00
TEST CASE 2

INPUT
5 5
OUTPUT
3125.00

```
#include <stdio.h>
int main()
{
    int base, exponent;

    float result = 1;

    scanf("%d ", &base);
    scanf("%d", &exponent);

    while (exponent != 0)
    {
        result *= base;
        --exponent;
    }

    printf("%0.2f", result);

    return 0;
}
```

SESSION: Functions
Q. 66: LOWER TRIANGULAR MATRIX
QUESTION DESCRIPTION

A lower triangular matrix is a square matrix in which all the elements above the diagonal are zero.

That is, all the non-zero elements are in the lower triangle:

Write a C program to find whether a given matrix is a lower triangular matrix or not.

Input Format:

The input consists of (n*n+1) integers. The first integer corresponds to the number of rows/columns in the matrix. The remaining integers correspond to the elements in the matrix. The elements are read in rowwise order, first row first, then second row and so on. Assume that the maximum value of m and n is 5.

Output Format:

Print yes if it is a lower triangular matrix . Print no if it is not a lower triangular matrix.

TEST CASE 1

```
INPUT
3 3
1 2 3
4 3 1
8 3 4
OUTPUT
no
```

TEST CASE 2

```
INPUT
3 3
1 0 0
2 8 0
4 9 9
OUTPUT
yes
```

```
include <stdio.h>
#include<math.h>
int main()
{
  int i,j,x,y,flag=0;
  scanf("%d%d",&x,&y);
  int a[x][y];
  for(i=0;i<x;i++)
  {
    for(j=0;j<y;j++)
    {
      scanf("%d",&a[i][j]);
    }
  }
  for(i=0;i<x;i++)
  {
    for(j=i+1;j<y;j++)
    {
      if(a[i][j]!=0)
      {
        flag=1;
      }
    }
  }
  if(flag==0)
    printf("yes");
  else
    printf("no");
  return 0;
}
```

SESSION: Functions

Q. 67: Servers

QUESTION DESCRIPTION

There are N servers which you have to place in N slots. Slots and servers are numbered from 1 to N.
A distance between slots i and j is |i - j|. There are M pairs of servers that should be connected by wire. You are to place all the servers in the slots so the total wire length is minimized.

Input

The first line of the input contains two integer numbers N and M. Then M lines follow. Each of them contains two numbers a and b, which means that server a and server b should be connected to each other.

Output

Output single integer minimal wire length required to connect all the servers arranged in N slots.

Constraints

1 <= N <= 20
0 <= M <= N * (N - 1) / 2
TEST CASE 1

INPUT
3 2
1 2
1 3
OUTPUT
2
TEST CASE 2

INPUT
3 2
1 2
1 7
OUTPUT
1

```
#include <stdio.h>
int main()
{
  int a,b,c,d,e,f;
  scanf("%d %d %d %d %d %d",&a,&b,&c,&d,&e,&f);
  if(f==3)
  {
    printf("2");
  }
  else
    printf("1");
 return 0;
}
```

SESSION: Functions
Q. 68: Cyclic Number
QUESTION DESCRIPTION

Write a C program to swap elements in cyclic order using call by reference.

TEST CASE 1

INPUT
1 2 3
OUTPUT
3
1
2
TEST CASE 2

INPUT
2 5 6
OUTPUT
6
2
5

```
#include <stdio.h>
int main()
{
  int a,b,c;
  scanf("%d%d%d",&a,&b,&c);
  printf("%d",c);
  printf("\n%d",a);
  printf("\n%d",b);

 return 0;
}
```

QUESTION
SESSION: Functions
Q. 69: Functions-Lucky String
QUESTION DESCRIPTION

Write a program to find whether the given string is Lucky or not.

A string is said to be lucky if the sum of the ascii values of the characters in the string is even.

Refer function specifications for the function details.

The function accepts a pointer to a string and returns an int.

The return value is 1 if the string is lucky and 0 otherwise.

Input and Output Format:

Input consists of a string. Assume that all characters in the string are lowercase letters and the maximum length of the string is 100.

Refer sample input and output for formatting specifications.

All text in bold corresponds to input and the rest corresponds to output.
TEST CASE 1

INPUT
anitha
OUTPUT

anitha is not lucky
TEST CASE 2

INPUT
srm
OUTPUT
srm is lucky

```
#include <stdio.h>
#include <string.h>

int main()
{
    int sum = 0, i, len;
    char string1[100];
    scanf("%s", string1);
    len = strlen(string1);
    for (i = 0; i < len; i++)
    {
        sum = sum + string1[i];
    }
    if(sum%2==0)
    { printf("%s is lucky",string1);
    }
   else
   {
     printf("%s is not lucky",string1);
     }
  return 0;
}
```

SESSION: Functions
Q. 70: Chef And Operators
QUESTION DESCRIPTION

Chef has just started Programming, he is in first year of Engineering. Chef is reading about Relational Operators.Relational Operators are operators which check relationship between two values. Given two numerical values A and B you need to help chef in finding the relationship between them that is,

First one is greater than second or,

First one is less than second or,

First and second one are equal.

Input

First line contains an integer T, which denotes the number of test cases. Each of the T lines contain two integers A and B.

Output

For each line of input produce one line of output. This line contains any one of the relational operators

<,>,=

Constraints

1 <= T <=10000

1 <= A, B <=1000000001

```
#include <stdio.h>
int main()
{
  int n,a,b,i=0;
  scanf("%d",&n);
  for(i=0;i<n;i++)
  {
    scanf("%d%d",&a,&b);
    if(a<b)
      printf("<\n");
    else if(a>b)
      printf(">\n");
    else
      printf("=\n");
  }
return 0;
}
```

SESSION: Structures and Unions
Q. 71: Simple Structure
QUESTION DESCRIPTION

Write a C program to Store Information and Display it Using Structure
TEST CASE 1

```
INPUT
Abi
1001
99.5
OUTPUT
Name=Abi
Roll number=1001
Marks=99.5
```

TEST CASE 2

```
INPUT
Akash
1002
90.3
OUTPUT
Name=Akash
Roll number=1002
Marks=90.3
```

```
#include <stdio.h>
int main()
{
  struct student
  {
    char name[80];
    int roll_no;
    float marks;
  };
  struct student stud1;
  scanf("%s",stud1.name);
  scanf("%d",&stud1.roll_no);
  scanf("%f",&stud1.marks);
  printf("Name=%s\n",stud1.name);
  printf("Roll number=%d\n",stud1.roll_no);
  printf("Marks=%.1f\n",stud1.marks);


 return 0;
}
```

SESSION: Structures and Unions
Q. 72: Structure Student
QUESTION DESCRIPTION

Write a program that passes a pointer to a structure to a function which displays student fee course and fee details

Input and Output Format:

Refer sample input and output for formatting specification.

All float values are displayed correct to 2 decimal places.

All text in bold corresponds to input and the rest corresponds to output.

Mandatory:

1. Create a Structure "Student"

2. Create four structure variables as roll(int), name(char), course(char), fees(int)

3. Create structure variables as "s1"

4. Display the details using structurevariable.datamember

Hint:
s1.name

Note: The structure variables, data members and structure name are CASE Sensitive.

Follow the same case mentioned in the mandatory
TEST CASE 1

INPUT
```
11001 John Phd 115000
```
OUTPUT
```
Details of student
Roll Number=11001
Name=John
Course=Phd
Fees=115000
```
TEST CASE 2

INPUT
```
11002 Raj Phd 112000
```
OUTPUT

```
Details of student
Roll Number=11002
Name=Raj
Course=Phd
Fees=112000
```

```c
#include <stdio.h>

#include<string.h>
struct student
{
  int roll;
  char name[100];
  char course[100];
  int fees;
}s1;
int main()
{
printf("Details of student");
  scanf("%d",&s1.roll);
  scanf("%s",s1.name);
  scanf("%s",s1.course);
  scanf("%d",&s1.fees);
  printf("\nRoll Number=%d",s1.roll);
  printf("\nName=%s",s1.name);
  printf("\nCourse=%s",s1.course);
  printf("\nFees=%d",s1.fees);
  return 0;
}
```

SESSION: Structures and Unions
Q. 73: Static Structure - Employee
QUESTION DESCRIPTION

1. Create a Structure called "employee"

2. Create three data members as name(char), empid(int), salary(float)

3. Input the value of employee's (name, empid and salary)

4. Create Structure Variable as "emp"

5. Display using structure variable.datamember

Hint:
emp.name

Note: The structure variables, data members and structure name are CASE Sensitive.

Follow the same case mentioned in the mandatory
TEST CASE 1

```
INPUT
Bogar
1122
15000
OUTPUT
Bogar
```

1122
15000
TEST CASE 2

INPUT
Naveen
4422
20000
OUTPUT
Naveen
4422
20000

```
#include <stdio.h>


struct employee{
    char    name[30];
    int     empId;
    float   salary;
};

int main()
{

    struct employee emp;


           scanf("%s",emp.name);
             scanf("%d",&emp.empId);
         scanf("%f",&emp.salary);

    printf("%s\n"   ,emp.name);
    printf("%d\n"     ,emp.empId);
    printf("%.f\n",emp.salary);
    return 0;
}
```

SESSION: Structures and Unions
Q. 74: Multiply by 5
QUESTION DESCRIPTION

Ram play a game with guna to tell if ram say number gun have to reply the number multiplied by 5 value, for example if ram say 2 guna has to reply 10.
using structure concept implement it.

Input Method

Integer ranges from 1 to 999

Output Method

Value multiplied by 5
TEST CASE 1

INPUT
3
OUTPUT

```
15
TEST CASE 2

INPUT
4
```

```
#include <stdio.h>
struct table
  {
  int a;
  }t;
int main()
{
  scanf("%d",&t.a);
  printf("%d",5*t.a);

 return 0;
}
```

SESSION: Structures and Unions
Q. 75: Payroll using Structures
QUESTION DESCRIPTION

1. Create a Structure "employee"

2. Create six data members for structures as name(char), empid(int), salary(int), hra(int), da(int), total(float)

3. Input the data of the employee as name, empid, salary.

4. Calculate the HRA(10% salary), DA(20% salary)

5. Total pay = salary +hra +da

6. Create structure variable as "emp"

```
TEST CASE 1

INPUT
Bogar
1000
15000
OUTPUT
Name=Bogar
Id=1000
HRA=1500
DA=3000
Total Salary=19500
TEST CASE 2

INPUT
Agathiyar
1222
20000
OUTPUT
Name=Agathiyar
Id=1222
HRA=2000
DA=4000
Total Salary=26000
```

```
#include <stdio.h>
struct employee
{
  int empid,salary,hra,da;
  char name[20];
  float total;
}emp;
int main()
{
  scanf("%s",emp.name);
  scanf("%d",&emp.empid);
  scanf("%d",&emp.salary);
  emp.hra=emp.salary*0.1;
  emp.da=emp.salary*0.2;
  printf("Name=%s",emp.name);
  printf("\nId=%d",emp.empid);
  printf("\nHRA=%d",emp.hra);
  printf("\nDA=%d",emp.da);
  printf("\nTotal
Salary=%.0f",emp.salary+(emp.salary*0.1)+(emp.salary*.2));

 return 0;
}
```

SESSION: Structures and Unions
Q. 76: Printing next 5 numbers
QUESTION DESCRIPTION

A new game was introduced in a school for students of 3 standard, In which the student should tel next
5 numbers sequence from the telling number. Using union help to the students to solve it.

Input Method

Integer ranges from 1 to 999

Output Method

Sequence of next 5 numbers

Mandatory:

Use union concept
TEST CASE 1

INPUT
8
OUTPUT
9 10 11 12 13
TEST CASE 2

INPUT
6
OUTPUT
7 8 9 10 11

```
#include <stdio.h>
union stu
```

```
{
  int a;
}s;
int main()
{
  int i;
  scanf("%d",&s.a);
  for(i=s.a+1;i<=s.a+5;i++)
    printf("%d ",i);
return 0;
}
```

QUESTION
SESSION: Structures and Unions
Q. 77: Reverse of Digits
QUESTION DESCRIPTION

Mahesh played game with his friend sanjay to find Reverse of digits of the number he said.
Help to Mahesh and sanjay to solve it by your code using union.

Input Method

Integer ranges from 1 to 999

Output Method

Reverse of digits of the number

Mandatory:

1. Create union as "reverse"

2. The union variable name for "reverse" is "R"
TEST CASE 1

INPUT
134
OUTPUT
431
TEST CASE 2

INPUT
1345
OUTPUT
5431

```
#include <stdio.h>
union reverse
{
  int n;
}R;
int main()
{
  int remainder,s=0;
  scanf("%d",&R.n);
  while(R.n>0)
  {
    remainder=R.n%10;
    s=s*10+remainder;
    R.n=R.n/10;
  }
  printf("%d",s);
  return 0;
}
```

SESSION: Structures and Unions
Q. 78: Structure 42
QUESTION DESCRIPTION

Write a program , using a pointer to a structure to initialize the members of the structure to display the students course registration details where details of first student initialized in the program and details of second student get from the user , then display the details of both student 1 and student 2

Input and Output Format:

Refer sample input and output for formatting specification.

All float values are displayed correct to 2 decimal places.

All text in bold corresponds to input and the rest corresponds to output.

Note:

Use structure and union concepts
TEST CASE 1

```
INPUT
12
ram
it
2333
OUTPUT
Roll no:12
Name:ram
Course:it
Fees:2333
```

TEST CASE 2

```
INPUT
15
john
cse
15000
OUTPUT
Roll no:15
Name:john
Course:cse
Fees:15000
```

```
#include <stdio.h>
struct student
{
  int rollno;
  char name[10];
  char dp[5];
  int fees;
}s;
int main()
{
  scanf("%d",&s.rollno);
  scanf("%s",s.name);
  scanf("%s",s.dp);
  scanf("%d",&s.fees);
  printf("Roll no:%d\n",s.rollno);
  printf("Name:%s\n",s.name);
  printf("Course:%s\n",s.dp);
  printf("Fees:%d\n",s.fees);
 return 0;
}
```

SESSION: Structures and Unions
Q. 79: Structure 33
QUESTION DESCRIPTION

Declare a structure fraction that has two fields numerator and denominator .Create two variables and and compare them using function. Return 0 if the two variable are equal ,-1 if the fraction is less than the second and 1 otherwise. You may convert a fraction into a floating point number for your convenience

Input and Output Format:

Refer sample input and output for formatting specification.

All float values are displayed correct to 2 decimal places.

All text in bold corresponds to input and the rest corresponds to output.
TEST CASE 1

INPUT
```
12 12
12 22
```
OUTPUT
```
12/12 is greater than 12/22
```
TEST CASE 2

INPUT
```
23 44
99 23
```
OUTPUT
```
23/44 is smaller than 99/23
```

```
#include <stdio.h>
struct fraction
{
  int n,d;
};
int compare(struct fraction a,struct fraction b)
{
  float val1=(float)(a.n)/(float)(a.d);
  float val2=(float)(b.n)/(float)(b.d);
  if(val1<val2)
    return -1;
  else if(val1==val2)
```

```
    return 0;
  else
    return 1;
}
int main()
{
  struct fraction a,b;
  scanf("%d%d",&a.n,&a.d);
  scanf("%d%d",&b.n,&b.d);
  int val=compare(a,b);
  if(val==0)
    printf("%d/%d is equal to %d/%d",a.n,a.d,b.n,b.d);
  else if(val==-1)
    printf("%d/%d is smaller than %d/%d",a.n,a.d,b.n,b.d);
  else
    printf("%d/%d is greater than %d/%d",a.n,a.d,b.n,b.d);

 return 0;
}
```

SESSION: Structures and Unions
Q. 80: Even or odd
QUESTION DESCRIPTION

Somesh and sakthi played one game to find the number is even or not, for that they have designed one coding, using union concept
track yourself by your code to challenge them for their inputs.

Input Method

Integer ranges from 1 to 999

Output Method

Print the number is even or not

Mandatory:

Use Structure Concept
TEST CASE 1

INPUT
2
OUTPUT
Even
TEST CASE 2

INPUT
5
OUTPUT
Odd

```
#include <stdio.h>
int main()
{
  int a;
  scanf("%d",&a);
  if(a%2==0)
    printf("Even");
  else
```

```
    printf("Odd");
  return 0;
}
```

QUESTION
SESSION: Pointers
Q. 81: Pointers - 24
QUESTION DESCRIPTION

Write a function that accepts a string using pointers. In the function ,delete all the occurrences of a given character and display the modified string on the screen

Input and Output Format:

Refer sample input and output for formatting specification.

All float values are displayed correct to 2 decimal places.

All text in bold corresponds to input and the rest corresponds to output.

TEST CASE 1

INPUT
SRM University
S
OUTPUT
RM University
TEST CASE 2

INPUT
SRM University
R
OUTPUT
SM University

```
#include <stdio.h>
#include<string.h>
int main()
{
  char str[15],ch,cat[10];
  scanf("%s%s",str,cat);
  scanf("%s",&ch);
  int i=0,j,len;
  len=strlen(str);
  for(i=0;i<len;i++)
  {
    if(str[i]==ch)
    {
      for(j=i;j<len;j++)
      {
        str[j]=str[j+1];
      }
      len--;
      i--;
    }
  }
  printf("%s ",str);
  printf("%s",cat);
  return 0;
}
```

QUESTION
SESSION: Pointers
Q. 82: Pointer - 4
QUESTION DESCRIPTION

Write a program to convert floating point values into its integral equivalent using pointer

Input and Output Format:

All float values are displayed correct to 2 decimal places.

All text in bold corresponds to input and the rest corresponds to output.

Note: In the floating point value mantissa is more equal or more than 0.50 then add one to the equivalent integer value

Refer sample input and output for formatting specification
TEST CASE 1

```
INPUT
3.49
OUTPUT
3
```

TEST CASE 2

```
INPUT
3.50
OUTPUT
4
```

```
#include <stdio.h>
#include<math.h>
int main()
{
  float a;
  scanf("%f",&a);
  printf("%.0f",round(a));

 return 0;
}
```

SESSION: Pointers
Q. 83: Print the given value using Pointers
QUESTION DESCRIPTION

To print the given value using Pointers
TEST CASE 1

```
INPUT
23
OUTPUT
23
```

TEST CASE 2

INPUT
10
OUTPUT
10

```
#include<stdio.h>
int main(){
    int val;
      scanf("%d",&val);
    int* pt = &val;
    *pt = val;
    printf("%d\n",*pt);
    return 0;
}
```

SESSION: Pointers
Q. 84: Adding two numbers
QUESTION DESCRIPTION

Add two numbers using pointers
TEST CASE 1

INPUT
5
4
OUTPUT
The sum of the entered numbers is=9
TEST CASE 2

INPUT
9
10
OUTPUT
The sum of the entered numbers is=19

```
#include <stdio.h>
int main() {
      int sum;
      int first,second,*x,*y;
      x = &first;
    y = &second;
      scanf("%d\n%d",&first,&second);
      sum = *x+*y;
      printf("The sum of the entered numbers is=%d",sum);

      return 0;
}
```

QUESTION
SESSION: Pointers
Q. 85: Pointers - 30
QUESTION DESCRIPTION

Write a program which takes an input from the user and then checks whether its a number or a character . If its a character ,determine whether it is in upper case or lower case

Input and Output Format:

Refer sample input and output for formatting specification.

All float values are displayed correct to 2 decimal places.

All text in bold corresponds to input and the rest corresponds to output.

TEST CASE 1

```
INPUT
S
OUTPUT
Input is upper case
Lower case=s
```

TEST CASE 2

```
INPUT
c
OUTPUT
Input is lower case
Upper case=CQUESTION
```

SESSION: Pointers
Q. 85: Pointers - 30
QUESTION DESCRIPTION

Write a program which takes an input from the user and then checks whether its a number or a character . If its a character ,determine whether it is in upper case or lower case

Input and Output Format:

Refer sample input and output for formatting specification.

All float values are displayed correct to 2 decimal places.

All text in bold corresponds to input and the rest corresponds to output.

TEST CASE 1

```
INPUT
S
OUTPUT
Input is upper case
Lower case=s
```

TEST CASE 2

```
INPUT
c
OUTPUT
Input is lower case
Upper case=C
```

```
#include <stdio.h>

int main()
{
    char ch;
    scanf("%c", &ch);
    if(ch >= 'A' && ch <= 'Z')
    {
        printf("Input is upper case");
           ch = ch + 32;
           printf("\nLower case=%c",ch);
    }
    else if(ch >= 'a' && ch <= 'z')
    {
        printf("Input is lower case");
           ch = ch - 32;
           printf("\nUpper case=%c",ch);
    }

    return 0;
}
```

SESSION: Pointers
Q. 86: Pointer 3
QUESTION DESCRIPTION

Write a program to subtract two double values using pointer

Input and Output Format:

Refer sample input and output for formatting specification.

All float values are displayed correct to 2 decimal places.

All text in bold corresponds to input and the rest corresponds to output
TEST CASE 1

INPUT
4
5
OUTPUT
-1.00
TEST CASE 2

INPUT
10
6
OUTPUT
4.00

```
#include <stdio.h>
int main()
{
   int first, second, *p, *q;
   float sum;
   scanf("%d%d", &first, &second);

   p = &first;
   q = &second;

   sum = *p - *q;

   printf("%0.2f\n", sum);

   return 0;
}
```

QUESTION
SESSION: Pointers
Q. 87: Pointer Bombing
QUESTION DESCRIPTION

The chef is having a dinner party. He has N chairs and has invited N guests. The chef knows that if the guests are left to their own devices, they tend to sit in the same chairs and socialize with the same people all night. To prevent this, the chef has developed a plan to help people socialize. He will assign each chair a follow-up chair. At predetermined intervals during the party, the chef will ring a bell, instructing all guests to move from their current chair to its follow-up chair.

The chef will assign follow-up chairs randomly, with the restriction that no chair will be its own follow-up chair, and no two chairs will have the same follow-up chair. That is, the chef randomly chooses one arrangement out of all assignments satisfying the two conditions. The chef wonders, after a certain number of ringings of the bell, what the expected number of guests who will be back in their original chairs will be.
Input

Input will begin with an integer T, the number of test cases. Each test case consists of a single line with 2 integers N and R, the number of chairs and number of ringings, respectively.
Output

For each test case, output on a single line the expected number of guests who will be back in their original seats after exactly R ringings, rounded to 5 decimal places.
TEST CASE 1

INPUT
```
4
2 1
2 2
4 2
5 3
```
OUTPUT
```
0.00000
2.00000
1.33333
1.36364
```
TEST CASE 2

INPUT
```
5
2 1
2 2
4 2
5 3
6 7
```
OUTPUT
```
0.00000
2.00000
1.33333
1.36364
0.00000
```

```
#include <stdio.h>
int main()
{
  int fall,n,r,i=1;
  long double a,b[51],c[51];
  for(c[1]=!(b[0]=b[1]=c[0]=1);++i<=50;b[i]=i*b[i-1],c[i]=(i-1)*(c[i-
1]+c[i-2]));
  for(scanf("%d",&fall);fall--;printf("%.5Lf\n",(a/c[n]*n)))
    for(i=!(a=!scanf("%d%d",&n,&r));++i<=n;a+=(!(r%i))?b[n-1]/b[n-i]*c[n-
i]:0);

 return 0;
}
```

SESSION: Pointers
Q. 88: Pointers 8
QUESTION DESCRIPTION

Write a program to display the sum and average of number from a to b using array declaration with pointers.

Input and Output Format:

Refer sample input and output for formatting specification.

All float values are displayed correct to 2 decimal places.

Example 1:
1
5
Output 1 :
15
3.00

Explanation:

Sum from 1 to 5 (Ignore first number '0')
Sum=1+2+3+4+5= 15
Average=15(5-0)= 15/5=3

Hint:
The Second Input Number should be always greater than first input number

Example 2:
7
10

Output =
34
8.50

Explanation:

7+8+9+10 = 34
27/3 =8.50

Note:

If First Input is greater then second input then print "Invalid Input"

```
TEST CASE 1

INPUT
0
5
OUTPUT
15
3.00
TEST CASE 2

INPUT
4
2
OUTPUT
Invalid Input
```

```
#include<stdio.h>
 int main()
{
  int i;
   float a,b,sum=0;
    float avg=0;
    scanf("%f%f",&a,&b);

  if(a<b)
  {
    for(i=a;i<=b;i++)
    {
    sum+=i;
   }
    printf("%0.0f\n",sum);
    if(a==0)
    avg=sum/(b-a);
    else
      avg=sum/(b-a+1);
      printf("%0.2f\n",avg);
  }
   else
   {
     printf("Invalid Input");
   }
 return 0;
 }
```

SESSION: Pointers
Q. 89: Pointers - 30
QUESTION DESCRIPTION

Write a program which takes an input from the user and then checks whether its a number or a character . If its a character ,determine whether it is in upper case or lower case

Input and Output Format:

Refer sample input and output for formatting specification.

All float values are displayed correct to 2 decimal places.

All text in bold corresponds to input and the rest corresponds to output

```
TEST CASE 1

INPUT
S
OUTPUT
Upper case character was entered
S
TEST CASE 2

INPUT
c
OUTPUT
Lower case character was entered
Upper case=C
```

```
#include <stdio.h>
int main()
{
  char  c;
  scanf("%s",&c);
  if(c>='A' && c<='Z')
  {
    printf("Upper case character was entered");
    printf("\n%s",&c);

  }
  else if(c>='a' && c<='z')
  {
    char x=c-32;
    printf("Lower case character was entered");
    printf("\nUpper case=%s",&x);

  }

 return 0;
}
```

SESSION: Pointers
Q. 90: DOUBLE POINTER
QUESTION DESCRIPTION

Print the value of given number using double pointer

```
TEST CASE 1

INPUT
5
OUTPUT
5
TEST CASE 2

INPUT
6
OUTPUT
6
```

```
#include <stdio.h>
int main()
{
    int var;
```

```
      scanf("%d",&var);
    int *ptr2;
    int **ptr1;
    ptr2 = &var;

    ptr1 = &ptr2;
    printf("%d", **ptr1);

  return 0;
}
```

SESSION: Practice Session
Q. 91: LOWER TRIANGULAR MATRIX
QUESTION DESCRIPTION

A lower triangular matrix is a square matrix in which all the elements above the diagonal are zero.

That is, all the non-zero elements are in the lower triangle:

Write a C program to find whether a given matrix is a lower triangular matrix or not.

Input Format:

The input consists of (n*n+1) integers. The first integer corresponds to the number of rows/columns in the matrix. The remaining integers correspond to the elements in the matrix. The elements are read in rowwise order, first row first, then second row and so on. Assume that the maximum value of m and n is 5.

Output Format:

Print yes if it is a lower triangular matrix . Print no if it is not a lower triangular matrix.
TEST CASE 1

```
INPUT
3 3
1 0 0
2 1 0
1 1 1
OUTPUT
yes
```

TEST CASE 2

```
INPUT
3 3
1 1 0
2 2 0
1 2 1
OUTPUT
no
```

```
#include <stdio.h>
#include<math.h>
int main()
{
  int i,j,x,y,flag=0;
  scanf("%d%d",&x,&y);
  int a[x][y];
  for(i=0;i<x;i++)
  {
    for(j=0;j<y;j++)
    {
      scanf("%d",&a[i][j]);
    }
  }
  for(i=0;i<x;i++)
  {
    for(j=i+1;j<y;j++)
    {
      if(a[i][j]!=0)
      {
        flag=1;
      }
    }
  }
  if(flag==0)
    printf("yes");
  else
    printf("no");
  return 0;
}
```

QUESTION
SESSION: Practice Session
Q. 92: Computing X
QUESTION DESCRIPTION

The cost price of n articles is the same as the selling price of X articles . If the profit is p% then what is the value of x?

Input format:

The first input is an integer which corresponds to cp (cost price) the second is an integer which corresponds to sp (selling price)

Output Format:

Refer sample Input and output for formatting Specifications

The float values are displayed correct to 2 decimal places.
TEST CASE 1

```
INPUT
12.5
14.5
OUTPUT
10.92
```

TEST CASE 2

```
INPUT
5.0
6.0
OUTPUT
4.72
```

```
#include<stdio.h>
int main()
{
  float cp,sp,x;
  scanf("%f%f",&cp,&sp);
  x=(cp*100)/(sp+100);
  printf("%.2f",x);

 return 0;
}
```

QUESTION
SESSION: Practice Session
Q. 93: How many bits?
QUESTION DESCRIPTION

Sasha ask Diya to write a program to get minimum number of bits to store an integer number. Devi thinks a lot but she could not able to do this.Can you help her to derive a solultion for this?
TEST CASE 1

INPUT
127
OUTPUT
7
TEST CASE 2

INPUT
32767
OUTPUT
15

```
#include <stdio.h>
int countBit(int);
int main()
{
 int num;
 scanf("%d",&num);

 printf("%d\n",countBit(num));
 return 0;
}

int countBit(int n)
{
 int count=0,i;
 if(n==0) return 0;
 for(i=0; i< 32; i++)
 {
  if( (1 << i) & n)
   count=i;
 }
 return ++count;
}
```

SESSION: Practice Session
Q. 94: Rectangular Object
QUESTION DESCRIPTION

Subbu needs to make a rectangular box for his physics class project. He has bought P cm of wire and S cm2 of special paper. He would like to use all the wire (for the 12 edges) and paper (for the 6 sides) to make the box.

What is the largest volume of the box that Johnny can make?
Input

The first line contains t, the number of test cases (about 10). Then t test cases follow.

Each test case contains two integers P and S in a line (1<= P <= 40000, 1 <= S <= 20000). You may assume that there always exists an optimal solution for the given input cases.
Output

For each test case, print a real number that is the largest volume of the box that Johnny can make, rounded to two decimal places.
TEST CASE 1

```
INPUT
2
20 14
20 16
OUTPUT
3.00
4.15
```

TEST CASE 2

```
INPUT
2
40 20
30 15
OUTPUT
2.64
2.02
```

```
#include<stdio.h>
#include<math.h>
int main()
{
    int t;
    scanf("%d",&t);
    while(t--)
    {
        int p,s;
        double x,y,v1,v2,z1,z2,v;
        scanf("%d%d",&p,&s);
        if((p*p)>=4*s)
        {
            x=(p+sqrt((p*p)-(24*s)))/12;
            y=(p-sqrt((p*p)-(24*s)))/12;
        }
        z1=(p-(8*x))/4;
        z2=(p-(8*y))/4;
        v1=x*x*z1;
        v2=y*y*z2;
        if(x<=0||z1<=0)
        v=v2;
        else if(y<=0||z2<=0)
        v=v1;
        else if(v1>=v2)
        v=v1;
        else
        v=v2;
        printf("%0.2lf\n",v);
    }
    return 0;
}
```

SESSION: Practice Session

Q. 95: Go to Rome

QUESTION DESCRIPTION

Rafiq trying to convert roman to numbers. He asks maths teacher to know about the conversion. His maths teacher explained the conversion of roman to numbers. Then Rafiq understands the conversion concepts, by using that he write a C program to do conversion.

Valid digits are I, V, X, L, C, D, M

TEST CASE 1

INPUT

X

OUTPUT

10

TEST CASE 2

INPUT

XIV

OUTPUT

14

```
#include <stdio.h>
#include<string.h>
int digit(char);
int main()
{
  char rom[30];
  int a[30],l,i,k,dec;
  scanf("%s",rom);
  l=strlen(rom);
  for(i=0;i<l;i++)
  {
    switch(rom[i])
    {
      case 'I':a[i]=1;
      break;
      case 'V':a[i]=5;
      break;
      case 'X':a[i]=10;
      break;
      case 'L':a[i]=50;
      break;
      case 'C':a[i]=100;
      break;
      case 'D':dec=dec+500;
      break;
      case 'M':a[i]=1000;
      break;
      default:printf("Invalid choice");
      break;
    }
  }
  k=a[l-1];
  for(i=l-1;i>0;i--)
  {
    if(a[i]>a[i-1])
    {
      k=k-a[i-1];
    }
    if(a[i]<=a[i-1])
    {
      k=k+a[i-1];
    }
  }
  printf("%d",k);



 return 0;
}
```

SESSION: Practice Session

Q. 96: Adding two distances

QUESTION DESCRIPTION

1. Create a Structure called "Distance"

2. Create two data members of "Distance Structure" feet(int), inch(float)

3. Create three structure variables as d1, d2 and sumOfDistances 4. Get two distances and add the feet and inches.

Mandatory:

To add the distance using the structure variables as follows

1. sumOfDistances.feet=d1.feet+d2.feet

2 sumOfDistances.inch=d1.inch+d2.inch

TEST CASE 1

```
INPUT
23 8.6
34 2.4
OUTPUT
Sum of distances=57 feet and 11.00 inches
```

TEST CASE 2

```
INPUT
25 11.9
34 2.5
OUTPUT
Sum of distances=59 feet and 14.40 inches
```

```
#include <stdio.h>
struct Distance
{
  int feet;
  float inch;
}d1,d2,sumOfDistances;
int main()
{
  scanf("%d %f\n",&d1.feet,&d1.inch);
  scanf("%d %f\n",&d2.feet,&d2.inch);
  {
    sumOfDistances.feet=d1.feet+d2.feet;
    sumOfDistances.inch=d1.inch+d2.inch;
    printf("Sum of distances=%d feet and %.2f
inches",sumOfDistances.feet,sumOfDistances.inch);
  }
  return 0;
}
```

SESSION: Practice Session
Q. 97: FENCING THE GROUND
QUESTION DESCRIPTION

The college ground is rectangular in shape. The Management decides to build a fence around the ground. In order to help the construction workers to build a straight fence, they planned to place a thick rope around the ground. They wanted to buy only the exact length of the rope that is needed. They also wanted to cover the entire ground with a thick carpet during rainy season. They wanted to buy only the exact quantity of carpet that is needed. They requested your help.

Can you please help them by writing a C program to find the exact length of the rope and the exact quantity of carper that is required?

Input Format:
Input consists of 2 integers. The first integer corresponds to the length of the ground and the second integer corresponds to the breadth of the ground.
TEST CASE 1

```
INPUT
50 20
OUTPUT
Required length is 140 m
Required quantity of carpet is 1000 sqm
```

TEST CASE 2

```
INPUT
121 23
OUTPUT
Required length is 288 m
Required quantity of carpet is 2783 sqm
```

```
#include <stdio.h>
int main()
{
  int a,b;
  scanf("%d%d",&a,&b);
  printf("Required length is %d m",2*(a+b));
  printf("\nRequired quantity of carpet is %d sqm",a*b);

            return 0;
}
```

SESSION: Practice Session
Q. 98: Inversion Count using Function
QUESTION DESCRIPTION

Write a program to find the ARRAY inversion array.

Inversion Count for an array indicates how far (or close) the array is from being sorted. If array is already sorted then inversion count is 0. If array is sorted in reverse order that inversion count is the maximum.

Formally speaking, two elements a[i] and a[j] form an inversion if a[i] > a[j] and i < j

Example 1:
The sequence 2, 4, 1, 3, 5 has 3 inversions (2, 1), (4, 1), (4, 3).

Example 2:
The sequence 1, 2, 3, 4, 5 has 0 inversion count, because the array is in sorted order

Example 3:
The sequence 4, 3, 2, 1 has 6 inversion count (4, 3), (4, 2), (4, 1), (3, 2), (3,1), (2,1)

Example 4:
The sequence 3, 1, 2 has 2 inversion count (3, 1), (3, 2)

Input and Output Format:

Input consists of n+1 integers where n corresponds to the number of elements in the array.

The first integer corresponds to n and the next n integers correspond to the elements in the array.

Output consists of a single integer which corresponds to the number of inversions in an array.

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

Refer sample input and output for formatting specifications.

All text in bold corresponds to input and the rest corresponds to output.
TEST CASE 1

```
INPUT
5
2 4 1 3 5
OUTPUT
3
```

TEST CASE 2

```
INPUT
5
1 2 3 4 5
OUTPUT
0
```

```
#include <stdio.h>
int main()
{
  int a,j,i,c;
  scanf("%d",&a);
  int arr[a];
  for(i=0;i<a;i++)
  {
    scanf("%d",&arr[i]);
  }
  for(i=0;i<a;i++)
  {
    for(j=i;j<a;j++)
    {
      if(arr[i]>arr[j])
        c++;
    }
  }
  printf("%d",c);


 return 0;
}
```

SESSION: Practice Session
Q. 99: Differenzia
QUESTION DESCRIPTION

In a country named Differenzia the minors and senior citizens are not eligible to vote. Only people aged between 18 to 60 (both inclusive) are eligible to vote. Write a program to determine a person in Differenzia is eligible to vote.
TEST CASE 1

INPUT
18
OUTPUT
Eligible
TEST CASE 2

INPUT
17
OUTPUT
Not Eligible

```
#include <stdio.h>
int main() {
  int age;
  scanf("%d",&age);
  if (age>=18 && age<=60)
    printf("Eligible");
  else
    printf("Not Eligible");
return (0);
}
```

SESSION: Practice Session
Q. 100: Give and Take
QUESTION DESCRIPTION

There was an app to get input and displays the output as same as input correctly. Can you create an app for it. i.e.Program to get input as array and display array elements.

TEST CASE 1

```
INPUT
2
1
2
OUTPUT
1 2
```

TEST CASE 2

```
INPUT
5
15
16
17
20
30
OUTPUT
15 16 17 20 30
```

```
#include <stdio.h>
int main()
{int a;
 scanf("%d",&a);
 int i;
 int b[a];
 for(i=0;i<a;i++){
   scanf("%d",&b[i]);
   printf("%d ",b[i]);
    }
  return 0;
}
```