**Section-1: Input output**

**Q-1 : Hello World**

We all know that the input and output functionality forms the basics of any programming language. As a universal practice, let us begin with this - Write a program to print "Hello World!".  
  
**Case 1**

**Input (stdin)**

Hello World!

**Output (stdout)**

Hello World!

#include<stdio.h>

int main()

{

  printf("Hello World!");

  return 0;

}

**Q-2 : Print the output with a new line**

Write a C program to print the below-given output.

**SAMPLE OUTPUT:**

Hello!

Welcome to Face!  
  
**Case 1**

**Input (stdin)**

Hello!  
Welcome to Face!

**Output (stdout)**

Hello!  
Welcome to Face!

#include<stdio.h>

int main()

{

  printf("Hello!\n");

  printf("Welcome to Face!");

  return 0;

}

**Q-3 : Integer**

Write a program to get an integer from the user and display it.

**INPUT & OUTPUT FORMAT:**

Input consist of 1 integer.

Output consist of 1 integer.  
  
**SAMPLE INPUT & OUTPUT:**

**Case 1**

**Input (stdin)**

56

**Output (stdout)**

56

**Case 2**

**Input (stdin)**

8

**Output (stdout)**

8

**Case 3**

**Input (stdin)**

28  
**Output (stdout)**

28

#include<stdio.h>

int main()

{

  int a;

  scanf("%d",&a);

  printf("%d",a);

}

**Q-4 : Floating Point**

Write a program to get a float value from the user and display it.  
  
**SAMPLE INPUT & OUTPUT 1:**

23.115

23.115000

**SAMPLE INPUT & OUTPUT 2:**

8

8.000000

**Case 1**

**Input (stdin)**

8  
**Output (stdout)**

8.000000

#include<stdio.h>

int main()

{

  float a;

  scanf("%f",&a);

  printf("%f",a);

}

**Q-5 : Character**

Write a program to get a character value from the user and display it.

**INPUT & OUTPUT FORMAT:**

Input consists of 1 character. Output consists of 1 character.  
  
**Case 1**

**Input (stdin)**

d

**Output (stdout)**

 d

**Case 2**

**Input (stdin)**

g

**Output (stdout)**

g

**Case 3**

**Input (stdin)**

t

**Output (stdout)**

t

#include<stdio.h>

int main()

{

  char a;

  scanf("%c",&a);

  printf("%c",a);

}

**Q-6 : Round Off**

Write a C program to get a float value from the user and display it in the below-mentioned format. HINT: Use ceil() and floor() functions from header file.

**INPUT & OUTPUT FORMAT:**

Input consists of one float value.

Output consists of one integer, its highest round off value and its lowest round off value.

**SAMPLE INPUT & OUTPUT:**

54.5

54

55.0

54.0

**Case 1**

**Input (stdin)**

54.5  
  
**Output (stdout)**

54  
55.0  
54.0

#include<stdio.h>

#include<math.h>

int main()

{

  float Val;

  float fVal,cVal;

  scanf("%f",&Val);

  cVal=ceil(Val);

  fVal=floor(Val);

  printf("%.f \n%.01f \n%.01f\n",Val,cVal,fVal);

  return 0;

}

**Q-7 : Print ASCII Values**

Write a C program to generate the equivalent ASCII value for a given character.

**SAMPLE INPUT & OUTPUT 1:**

A

65

**SAMPLE INPUT & OUTPUT 2:**

a

97

**Case 1**

**Input (stdin)**

A

**Output (stdout)**

65

**Case 2**

**Input (stdin)**

a

**Output (stdout)**

97

#include<stdio.h>

int main()

{

char a;

 scanf("%c",&a);

 printf("%d",a);

}

**Q-8 : Tsunami Report**

In Japan, there was a very huge Tsunami. Millions and millions worth buildings and property were destroyed. Many people lost their lives while some of them were injured and few were safe. A news reporter arrives at the spot to take the current survey regarding the situation of the people alive, dead and injured. He wanted to publish it in the newspaper and reach out to other countries asking to help the affected people. Can you please help him in this noble cause by writing a program to generate the newspaper report ?

**INPUT & OUTPUT FORMAT:** Input consists of three integers corresponding to the number of people dead, injured and those who are still alive and safe. Refer sample input and output for formatting specifications.

2000

3000

10000

**SAMPLE OUTPUT:**

1)Dead : 2000

2)Injured : 3000

3)Safe : 10000

Please help the people who are suffering!!!

**Case 1**

**Input (stdin)**

2000  
3000  
10000

**Output (stdout)**

1)Dead : 2000  
2)Injured : 3000  
3)Safe : 10000  
Please help the people who are suffering!!!

#include<stdio.h>

int main()

{

int Dead,Injured,Safe;

  scanf("%d",&Dead);

  scanf("%d",&Injured);

  scanf("%d",&Safe);

  printf("1)Dead : %d\n",Dead);

  printf("2)Injured : %d\n",Injured);

  printf("3)Safe : %d\n",Safe);

  printf("Please help the people who are suffering!!!");

}

**Section-2: Operations**

**Q-1 : Artihmetic operator**

Write a C program to perform arithmetic operations such as addition, subtraction, multiplication, modulo division and division.

**INPUT & OUTPUT FORMAT:**

Input consists of two integers Output consist of integers

**SAMPLE INPUT:**

5

2

**SAMPLE OUTPUT:**

7

3

10

1

2

**Case 1**

**Input (stdin)**

5  
2  
**Output (stdout)**

7  
3  
10  
1  
2

**Case-2**

**Input (stdin)**

9  
3

**Output (stdout)**

12  
6  
27  
0  
3

 #include<stdio.h>

int main()

{

int a,b,c,d,e,f,g;

  scanf("%d",&a);

  scanf("%d",&b);

  c=a+b;

  d=a-b;

  e=a\*b;

  f=a%b;

  g=a/b;

  printf("%d\n%d\n%d\n%d\n%d\n",c,d,e,f,g);

}

**Q-2 : Cricket Stadium**

There was a large ground in center of the city which is rectangular in shape. The Corporation decides to build a Cricket stadium in the area for school and college students, But the area was used as a car parking zone. In order to protect the land from using as an unauthorized parking zone , the corporation wanted to protect the stadium by building a fence. In order to help the workers to build a 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 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 program to find the exact length of the rope and the exact quantity of carpet 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.

**Output Format:**

Output Consists of two integers. The first integer corresponds to the length. The second integer corresponds to the quantity of carpet required.

**Sample Input:**

50

20

**Sample Output:**

140

1000

**Case 1**

**Input (stdin)**

50  
20

**Output (stdout)**

140  
1000

**Case-2**

**Input (stdin)**

70  
40

**Output (stdout)**

220  
2800

#include<stdio.h>

int main()

{

int l,w;

  scanf("%d",&l);

  scanf("%d",&w);

  printf("%d",(2\*l)+(2\*w));

  printf("\n%d",l\*w);

  return 0;

}

**Q-3 : Dept Repay**

Alice wanted to start a business and she was looking for a venture capitalist. Through her friend Bob, she met the owner of a construction company who is interested to invest in an emerging business. Looking at the business proposal, the owner was very much impressed with Alice's work. So he decided to invest in Alice's business and hence gave a green signal to go ahead with the project. Alice bought Rs.X for a period of Y years from the owner at R% interest per annum. Find the rate of interest and the total amount to be given by Alice to the owner. The owner impressed by proper repayment of the financed amount decides to give a special offer of 2% discount on the total interest at the end of the settlement. Find the amount given back by Alice and also find the total amount. (Note: All rupee values should be in two decimal points).

**INPUT FORMAT:**

Input consists of 3 integers. The first integer corresponds to the principal amount borrowed by Alice. The second integer corresponds to the rate of interest The third integer corresponds to the number of years.

**OUTPUT FORMAT:**

The output consists of 4 floating point values. The first value corresponds to the interest. The second corresponds to the amount. The third value corresponds to the discount. The last value corresponds to the final settlement. All floating point values are to be rounded off to two decimal places

**SAMPLE INPUT:**

100

1

10

**SAMPLE OUTPUT:**

10.00

110.00

0.20

109.80

**Case 1**

**Input (stdin)**

100  
1  
10  
**Output (stdout)**

10.00  
110.00  
0.20  
109.80

**Case-2**

**Input (stdin)**

10000  
10  
5  
**Output (stdout)**

5000.00  
15000.00  
100.00  
14900.00

#include<stdio.h>

int main()

{

 int x,y,z;

 float i,a,d,f;

 scanf("%d",&x);

 scanf("%d",&y);

 scanf("%d",&z);

 scanf("%f,%f,%f,%f",&i,&a,&d,&f);

 i=(x\*y\*z)/100;

 a=x+i;

 d=(i\*2)/100;

 f=a-d;

 printf("%.2f\n%.2f\n%.2f\n%.2f",i,a,d,f);

}

**Q-4 : Sports day Celebration**

Training for sports day has begun and the physical education teacher has decided to conduct some team games. The teacher wants to split the students in higher secondary into equal sized teams. In some cases, there may be some students who are left out from the teams and he wanted to use the left out students to assist him in conducting the team games. For instance, if there are 50 students in a class and if the class has to be divided into 7 equal sized teams, 7 students will be there in each team and 1 student will be left out. That 1 student will assist the PET. With this idea in mind, the PET wants your help to automate this team splitting task. Can you please help him out?

**INPUT FORMAT:** Input consists of 2 integers. The first integer corresponds to the number of students in the class and the second integer corresponds to the number of teams.

**OUTPUT FORMAT:** The output consists of two integers. The first integer corresponds to the number of students in each team and the second integer corresponds to the students who are left out.

**SAMPLE INPUT:**

60

8

**SAMPLE OUTPUT:**

7

4

**Case 1**

**Input (stdin)**

60  
8  
**Output (stdout)**

7  
4

**Case-2**

**Input (stdin)**

50  
4

**Output (stdout)**

12  
2

#include<stdio.h>

int main()

{

  int a,b;

  scanf("%d%d",&a,&b);

  printf("%d\n%d\n",a/b,a%b);

}

**Q-5 : The newspaper Agency**

Each Sunday, a newspaper agency sells w copies of a special edition newspaper for Rs.x per copy. The cost to the agency of each newspaper is Rs.y. The agency pays a fixed cost for storage, delivery and so on of Rs.100 per Sunday. The newspaper agency wants to calculate the profit which it obtains only on Sundays. Can you please help them out by writing a program to compute the profit if w, x, and y are given.

**INPUT FORMAT:** Input consists of 3 integers: w, x, and y. w is the number of copies sold, x is the cost per copy and y is the cost the agency spends per copy.

**OUTPUT FORMAT:** The output consists of a single integer which corresponds to the profit obtained by the newspaper agency.

**SAMPLE INPUT:**

1000

2

1

**SAMPLE OUTPUT:**

900

**Case 1**

**Input (stdin)**

1000  
2  
1

**Output (stdout)**

900

**Case-2**

**Input (stdin)**

900  
5  
3  
**Output (stdout)**

1700

#include<stdio.h>

int main()

{

  int a,b,c,d,n,p,q;

  scanf("%d%d%d",&a,&b,&c);

  q=a\*b;

  n=a\*c;

  p=q-n;

  d=p-100;

  printf("%d",d);

}

**Q-6 : The Chronicles of Narnia**

Four kids Peter,Susan,Edmond and Lucy travel through a wardrobe to the land of Narnia. Narnia is a fantasy world of magic with mythical beasts and talking animals.While exploring the land of narnia Lucy found Mr.Tumnus the two legged stag ,and she followed it, down a narrow path .She and Mr.Tumnus became friends and he offered a cup of coffee to Lucy in his small hut.It was time for Lucy to return to her family and so she bid good bye to Mr.Tumnus and while leaving Mr.Tumnus told that it is quite difficult to find the route back as it was already dark.He told her to see the trees while returning back and said that the first tree with two digits number will help her find the way and the way to go back to her home is the sum of digits of the tree and that numbered way will lead her to the tree next to the wardrobe where she can find the others.Lucy was already confused, so pls help her in finding the route to her home....

**Input Format:** Input consists of an integer corresponding to the 2-digit number.

**Output Format:** Output consists of an integer corresponding to the sum of its digits.

**SAMPLE INPUT :**

87

**SAMPLE OUTPUT:**

15

**Case 1**

**Input (stdin)**

54  
**Output (stdout)**

9

**Case-2**

**Input (stdin)**

87  
**Output (stdout)**

15

#include<stdio.h>

int main()

{

  int a,b,c;

  scanf("%d",&b);

  a=b%10;

  c=b/10;

  printf("%d",a+c);

  return 0;

}

**Q-7 : Four musketeers**

'Artagnan joined the group of 3 Musketeers and now their group is called four Musketeers. Meanwhile, d'Artagnan also moved to a new house in the same locality nearby to the other three. Currently, the houses of Athos, Porthos and Aramis are located in the shape of a triangle. When the three musketeers asked d'Artagnan about the location of his house, he said that his house is equidistant from the houses of the other 3. Can you please help them find out the location of the house? Given the 3 locations {(x1,y1), (x2,y2) and (x3,y3)} of a triangle, write a program to determine the point which is equidistant from all the 3 points.

**INPUT FORMAT:** Input consists of 6 integers. The first integer corresponds to x1. The second integer corresponds to y1. The third and fourth integers correspond to x2 and y2 respectively. The fifth and sixth integers correspond to x3 and y3 respectively.

**OUTPUT FORMAT:** The output consists of two floating point numbers (with one decimal place) which correspond to the location of the house.

**SAMPLE INPUT:**

2

4

10

15

5

8

**SAMPLE OUTPUT:**

5.7

9.0

**Case 1**

**Input (stdin)**

2  
4  
10  
15  
5  
8

**Output (stdout)**

5.7  
9.0

**Case-2**

**Input (stdin)**

1  
1  
1  
1  
1  
1  
1  
**Output (stdout)**

1.0  
1.0

#include<stdio.h>

int main()

{

  float x1,x2,x3,y1,y2,y3;

  scanf("%f%f%f%f%f%f",&x1,&y1,&x2,&y2,&x3,&y3);

  printf("%.1f\n%.1f",(x1+x2+x3)/3,(y1+y2+y3)/3);

  return 0;

}

**Q-8 : The misers discount**

Mrs.Bhulbhul is a miser to the core. She saves money even on petite things. One day she heard a discount offer announced in a mall. She wants to purchase a lot of items to save her money. The discount is given only when at least two items are bought. Since each item has different discount prices, she finds it difficult to check the amount she has saved. So she approaches you to device an automated discount calculator to make her easy while billing.

**INPUT FORMAT:** Input consists of two floating point values denoting price of item1 and item2. The third input denotes the discount value in percentage.

**OUTPUT FORMAT:** The output consists of three floating values denoting total amount, discounted price and amount saved.

**SAMPLE INPUT:**

20.50

45.40

10

**SAMPLE OUTPUT:**

65.90

59.31

6.59

**Case 1**

**Input (stdin)**

20.50  
45.40  
10  
**Output (stdout)**

65.90  
59.31  
6.59

**Case-2**

**Input (stdin)**

65.90  
59.31  
6.59

**Output (stdout)**

125.21  
116.96  
8.25

#include<stdio.h>

int main()

{

  float a,b,c,x,y,y1,z;

  scanf("%f%f%f",&a,&b,&c);

  x=a+b;

  printf("%.2f",x);

  y=x\*c/100;

  y1=x-y;

  printf("\n%.2f",y1);

  z=x\*c/100;

  printf("\n%.2f",z);

  return 0;

}

**Q-9 : Treasure Hunter**

Though there have been more successful pirates, Blackbeard is one of the best-known and widely-feared of his time. He commanded four ships and had a pirate army of 300 at the height of his career and defeated the famous warship, HMS “Scarborough” in sea-battle. He was known for barreling into battle clutching two swords with several knives and pistols at the ready. He captured over forty merchant ships in the Caribbean and without flinching killed many prisoners. Now, Blackbeard and his three pirates found a treasure of gold coins. Long Ben too joined them. They decided to share the treasure. Blackbeard agreed to give x% share for Long Ben. He then decided to take y% share from the remaining treasure. His other pirates will share the remaining gold coins equally. Write a program to compute their share's.

**INPUT FORMAT:** Input consists of 3 integers. The first input corresponds to the number of gold coins in the treasure. The second input corresponds to Ben's share percentage and the last input is Blackbeard's share percentage.

**OUTPUT FORMAT:** The output consists of three integers. The first output integer corresponds to Long Ben's share. The second integer corresponds to Blackbeard's share. The last integer corresponds to other pirates share.

**SAMPLE INPUT:**

729

65

87

**SAMPLE OUTPUT:**

473

222

11

**Case 1**

**Input (stdin)**

729  
65  
87  
**Output (stdout)**

473  
222  
11

**Case-2**

**Input (stdin)**

72  
6  
8  
**Output (stdout)**

4  
5  
21

#include<stdio.h>

int main()

{

  int a,b,c,x,y,z;

  scanf("%d%d%d",&a,&b,&c);

  x=(a\*b)/100;

  printf("%d\n",x);

  x=a-x;

  y=(x\*c)/100;

  printf("%d\n",y);

  z=x-y;

  printf("%d",z/3);

}

**Q-10 : Booka the alien**

Booka is an alien. He couldn't understand how to measure days, weeks, months and years. Make Booka understand what is meant by days, weeks, months and years. Teach him about the conversion of days into years, months and weeks using a program.

**INPUT FORMAT:** Input consists of an integer which corresponds to the number of days. **OUTPUT FORMAT:** The output consists of three integers. The first integer corresponds to the total years. The second integer corresponds to the total weeks. The third integer corresponds to the total days.

**SAMPLE INPUT:**

373

**SAMPLE OUTPUT:**

1

1

1

**Case 1**

**Input (stdin)**

373

**Output (stdout)**

1  
1  
1

**Case-2**

**Input (stdin)**

365  
**Output (stdout)**

1  
0  
0

#include<stdio.h>

int main()

{

 int ndays, year, week, days;

    scanf("%d", &ndays);

    year = ndays / 365;

    week =(ndays % 365) / 7;

    days =(ndays % 365) % 7;

    printf ("%d\n",year);

               printf ("%d\n",week);

    printf ("%d\n",days);

  return 0;

}

**Q-11 : Harry Potter**

Having crossed the three-headed dog, Harry, Ron, and Hermoine went through a secret trap door in search of Sorcerer's stone. On the way, they passed through a room and found that the room has only one door opposite to them and the door through which they entered has shut once they entered the room. The door was very large with a four digit number imprinted on it. When Harry and Ron tried to open it by casting out spells, it didn't open. Having tried various spells both of them got fed up and they left the task to Hermoine. Hermoine on curiously observing the room found that a statement was written on the top of the room. It was written as follows "I will be always four" “I can only be opened when you add my first and last and enter it” “If you find a sign, you should not consider it” Help Hermoine break the code and open the door so that they can save the sorcerer's stone.

**INPUT FORMAT:** Input consists of a single integer which is present on the door.

**OUTPUT FORMAT:** The output is a single integer.

**SAMPLE INPUT:**

1001

**SAMPLE OUTPUT:**

2

**Case 1**

**Input (stdin)**

1001

**Output (stdout)**

2

**Case-2**

**Input (stdin)**

2381  
**Output (stdout)**

3

#include<stdio.h>

int main()

{

  int a;

  scanf("%d",&a);

  int b=a/1000;

  int c=a%10;

  c+=b;

  printf("%d",c);

}

**Q-12 : Hop n Hop**

Big Bunny lives in a colony. He is the only bunny in his colony who is not able to hop. On his 5th birthday, his father bunny gifted him a pogo stick as he could not jump like the other bunnies. He is so excited to play with the pogo stick. The pogo stick hops one unit per jump. He wanders around his house jumping with pogo sticks. He wants to show the pogo stick to his friends and decides to go using pogo sticks. Write a program to find the number of hops needed to reach his friends' house. Assume that Big Bunny's house is in the location (3,4).

**INPUT FORMAT:** Input consists of two integers x,y. The x and y correspond to x and y coordinates of his friends' house.

**OUTPUT FORMAT:** The output is an integer. It corresponds to the number of hops needs to reach his friends' house.

**SAMPLE INPUT:**

5

10

**SAMPLE OUTPUT:**

6

**Case 1**

**Input (stdin)**

5  
10  
**Output (stdout)**

6

**Case-2**

**Input (stdin)**

15  
3  
**Output (stdout)**

12

#include<stdio.h>

int main()

{

  int x,y,x1,y1;

  scanf("%d%d",&x,&y);

  x1=x-3;

  y1=y-4;

  (x1>y1)? (printf("%d",x1)): (printf("%d",y1));

  return 0;

}

**Q-13 : Wind chill factor**

Wind chill factor is the felt air temperature on exposed skin due to the wind. The wind chill temperature is always lower than the air temperature and is calculated as per the following formula. WCF = 35.74 + 0.6215t + (0.4275t - 35.75) \* v^0.16 Write a program to receive values of temperature and wind velocity and calculate the Wind Chill Factor.

**INPUT FORMAT:** The input consists of two integers. The first integer corresponds to the temperature of the wind and the second integer corresponds to the wind velocity.

**OUTPUT FORMAT:** The output consists of a single integer which corresponds to the Wind Chill Factor. Decimal values are rounded off to two decimal points.

**SAMPLE INPUT:**

35

20

**SAMPLE OUTPUT:**

23.92

**Case 1**

**Input (stdin)**

35  
20  
**Output (stdout)**

23.92

**Case-2**

**Input (stdin)**

0  
0  
**Output (stdout)**

35.74

#include<stdio.h>

#include<math.h>

int main()

{

  int t,v;

  float w,n,k;

  scanf("%d%d",&t,&v);

  //n=(0.4275\*t)-35.75;

  //k=pow(v,0.16);

  w=35.74+0.6215\*t+((0.4275\*t)-35.75)\*pow(v,0.16);

  printf("%.2f",w);

}

**Decision Making Statements-I**

**Q-1 : Greatest of two numbers**

Write a C program to find the greatest among the given two numbers.

**INPUT FORMAT:** Input consist of 2 integers

**SAMPLE INPUT:**

77

99

**SAMPLE OUTPUT:**

99 is greater

**Input (stdin)**

5  
6  
**Output (stdout)**

6 is greater

#include<stdio.h>

int main()

{

  int a, b;

  scanf("%d %d", &a, &b);

  if(a>b)

    printf("%d is greater", a);

  else

    printf("%d is greater", b);

}

**Q-2 : Greatest of three numbers**

Write a C program to find the greatest among the three given numbers.

**INPUT FORMAT:** Input consists of 3 integers

**SAMPLE INPUT & OUTPUT**

7

9

5

9 is greater.

**Input (stdin)**

-1  
-33  
-45  
**Output (stdout)**

-1 is greater

#include<stdio.h>

int main()

{

  int a, b, c;

  scanf("%d %d %d", &a, &b, &c);

  if(a>b)

{

    if(a>c)

{

      printf("%d is greater", a);

               }

    else if(b>c)

{

      printf("%d is greater", b);

               }

    Else

{

      printf("%c is greater", c);

               }

  }

  else if(b>c)

{

    printf("%d is greater", b);

  }

  else

    printf("%d is greater", c);

}

**Q-3: Positive or Negative**

Write a C program to find whether the given number is positive or negative.

**INPUT FORMAT:** Input consists of 1 integer.

**SAMPLE INPUT & OUTPUT 1:**

56

Positive

**SAMPLE INPUT & OUTPUT 2:**

-56

Negative

**Input (stdin)**

56  
**Output (stdout)**

Positive

#include<stdio.h>

int main()

{

  int a;

  scanf("%d", &a);

  if(a>=0)

    printf("Positive");

  else

    printf("Negative");

}

**Q-4 : Quadrants**

Write a C program to find the quadrant in which the given x and y coordinates are present in a graph. If it is in 1st Quadrant, then display Ist Quadrant, If it is in 2nd Quadrant, then display IInd Quadrant, If it is in 3rd Quadrant, then display IIIrd Quadrant, If it is in 4th Quadrant, then display IVth Quadrant and If it is in center display Origin.

**INPUT FORMAT:** Input consists of 2 integers. First input corresponds to x coordinate. Second input corresponds to y coordinate.

**SAMPLE INPUT:**

5

6s

**SAMPLE OUTPUT:**

Ist Quadrant

Input (stdin)

-5  
6  
Output (stdout)

IInd Quadrant

#include<stdio.h>

int main()

{

  int a, b;

  scanf("%d %d", &a, &b);

  if(a>0 && b>0)

    printf("Ist Quadrant");

  else if(a<0 && b>0)

    printf("IInd Quadrant");

  else if(a>0 && b<0)

    printf("IIIrd Quadrant");

  else

    printf("IVth Quadrant");

}

Q-5 : Game with shapes

Karl loves to play with shapes and his sister is interested in Mathematics. They got a wooden box which is square in shape with length x cm and a round bangle with radius r cm and decided to play with it. They started playing and his sister asked him to find whether the circle will fit into the square. Help Karl to find whether the circle will fit into a square or not. If the circle fits, display “circle can be inside a Square” else display “circle cannot be inside a Square”. INPUT FORMAT: Input consist of 2 integers. First input corresponds to the radius of a circle. Second input corresponds to the length of a square.

SAMPLE INPUT:

77

65

**SAMPLE OUTPUT:**

circle cannot be inside a Square

**Input (stdin)**

28  
56  
**Output (stdout)**

circle can be inside a square

#include<stdio.h>

int main()

{

  int a, b;

  scanf("%d %d", &a, &b);

  if(a>b)

    printf("circle cannot be inside a Square");

  else

    printf("circle can be inside a square");

}

**Q-6 : Boating**

You went on a tour to Ooty with your friends. As a part of the tour, you went boating with them. For the boat to remain stable, the number of people on one boat is restricted based on the weight of the people. You find that the boatman who is sailing your boat is so much greedy of money. For earning more, he takes too many people to travel in the boat at a time. So you want to check how many people can travel in the boat at a time so that the boat will not drown. Calculate the weight by considering the number of adults and number of children. Assume that an adult weighs 75 kg and children weigh 30 kg each. If the weight is normal, display Boat is stable, else display Boat will drown.

**INPUT & OUTPUT FORMAT:** Input consists of 3 integers. First input corresponds to the weight that the boat can handle. Second input corresponds to the number of adults. Third input corresponds to the number of children.

**SAMPLE INPUT:**

340

2

3

**SAMPLE OUTPUT:**

Boat is stable

**Case:2**

**Input (stdin)**

600  
7  
4

**Output (stdout)**

Boat will drow

#include<stdio.h>

int main()

{

  int a, b, c, total;

  scanf("%d %d %d", &a, &b, &c);

 total = (b \* 75) + (c \* 30);

  if(total>a)

    printf("Boat will drow");

  else

    printf("Boat is stable");

}

**Q-7 : Car Race**

Janu is very much interested in watching car racing. Once while doing so, this is what she observes that Car 1 is coming so fast. The road on which the car comes is divided into three. There are three roads A, B, and C. Each road is given some number. Before the car reaches the end of the road, Janu wants to know the road in which the car will travel. Help her in finding in which road the car will travel?

**INPUT & OUTPUT FORMAT**: Input consists of 4 integers. First input corresponds to the number of road A. Second input corresponds to the number of road B. Third input corresponds to the number of road C. Fourth input corresponds to the number of cars. If the car cannot travel on any road, display “No path exists”.

**SAMPLE INPUT:**

243

136

567

17

**SAMPLE OUTPUT:**

Car 1 goes into road B

**Input (stdin)**

77  
55  
67  
9  
**Output (stdout)**

No path exist

#include<stdio.h>

int main()

{

  int a, b, c, d;

  scanf("%d %d %d %d", &a, &b, &c, &d);

  if (a%d==0)

    printf("Car 1 goes into road A");

  else if(b%d==0)

    printf("Car 1 goes into road B");

  else if(c%d==0)

    printf("Car 1 goes into road C");

  else

    printf("No path exist");

}

**Q-8 : Automorphic Number**

Janu is very interested in mathematics and she loves to solve math puzzles daily. Her friend knows that and she asks her to find out what an automorphic number is and to implement that concept using C programming. But since Janu is not good at programming, you need to help her implement this concept. An automorphic number is a number whose square ends with the number itself. For example, 5 is an automorphic number as 5\*5 =25. The last digit is 5 which is the same as the given number. It has only positive single digit numbers. If the number is not valid, it should display "Invalid input". If it is an automorphic number display “Automorphic Number” else display “Not Automorphic Number”.

**INPUT FORMAT:** Input consists of 1 integer.

**SAMPLE INPUT:**

7

**SAMPLE OUTPUT:**

Not Automorphic Number

**Input (stdin)**

6

**Output (stdout)**

Automorphic Number

#include<stdio.h>

int main()

{

  int a, sq;

  scanf("%d", &a);

  sq = a \* a;

  sq = sq % 10;

  if(a==sq)

    printf("Automorphic Number");

  else

    printf("Not Automorphic Number");

}

**Q-9 : Electricity Bill**

Electricity board has decided to charge rupees based on the units consumed by a particular home. If the units consumed is less than or equal to 200, the cost for one unit is 0.5. If the unit is less than or equal to 400, the cost for one unit is 0.65 and Rs.100 extra charge. If the unit is less than or equal to 600, the cost for one unit is 0.80 and Rs.200 extra charge. If the unit is greater than 600 the cost for one unit is 1.25 and Rs.425 extra charge. You need to now calculate the electricity bill based on the units consumed (given input).

**INPUT & OUTPUT FORMAT:** Input consists of one integer. This corresponds to the units consumed. The output consists of one integer. This corresponds to the electricity bill in rupees.

**SAMPLE INPUT:**

200

**SAMPLE OUTPUT:**

Rs.100

**Input (stdin)**

250

**Output (stdout)**

Rs.263

#include<stdio.h>

int main()

{

  int a;

  scanf("%d", &a);

  if(a<=200)

{

    a = a \* 0.5;

    printf("Rs.%d", a);

  }

  else if(a<=400)

{

    a = a \* 0.65;

    a = a + 101;

    printf("Rs.%d", a);

  }

  else if(a<=600)

{

    a = a \* 0.80;

    a = a + 200;

    printf("Rs.%d", a);

  }

  Else

{

    a = a \* 1.25;

    a = a + 425;

    printf("Rs.%d", a);

  }

}

**Q-10 : ASCII values** - Uppercase, Lowercase, Number or Symbol

Write a C program to find whether a given character is an uppercase, lowercase, number or symbol. If it is an alphabet find whether it is an Uppercase or Lowercase. If it is a number display Number else display Symbol.

**INPUT FORMAT:**

Input consists of 1 character.

**SAMPLE INPUT:**

A

**SAMPLE OUTPUT:**

Upper

**Input (stdin)**

^  
**Output (stdout)**

Symbol

#include<stdio.h>

int main()

{

  char a;

  scanf("%c", &a);

  if(a>='a' && a<='z')

    printf("Lower");

  else if(a>='A' && a<='Z')

    printf("Upper");

  else if(a>='0' && a<='9')

    printf("Number");

  else

    printf("Symbol");

}

**Q-11 : Eligibility for placements**

In your college, next week there is an on-campus drive happening for your seniors. The company has asked for the short-listed students names. The following are their criteria to become eligible for the campus interview. Students with 0 arrears and with 7.0 and above 7.0 CGPA Students with 1 or 2 arrears and with 7.5 and above 7.5 CPGA So your placement officer wants to shortlist the student names for the interview. As you are good at programming, he wants you to help him shortlist the student names. Help him by solving the program and display “Eligible to attend placement” or “Not Eligible to attend placement”.

**INPUT & OUTPUT FORMAT:**

Input consists of 1 string, 1 float, and 2 integers. The first input corresponds to the name of the student. Second input corresponds to register number. Third input corresponds to his/her CPGA. Fourth input corresponds to the number of arrears. The output consists of 2 strings and 1 integer.

**SAMPLE INPUT:**

Jai

69

7.3

2

**SAMPLE OUTPUT:**

Jai

69

Not Eligible to attend placement

#include<stdio.h>

int main(){

  char name[20];

  int num, arrears;

  float cpga;

  scanf("%s %d %f %d", &name, &num, &cpga, &arrears);

  printf("%s\n", name);

  printf("%d\n", num);

  if(arrears == 0 && cpga >= 7.0){

    printf("Eligible to attend placement"); }

  else if((arrears == 1 || arrears == 2) && cpga >=7.5)

    printf("Eligible to attend placement");

  else

    printf("Not Eligible to attend placement");

}

**Q-12 : Gift for Birthday**

Lucy is celebrating her 15th birthday. Her father promised her that he will buy her a new computer on her birthday if she solves the question asked by him. He asks Lucy to find out whether the year on which she was born is a leap year or not. Help her to solve this puzzle so that she celebrates her birthday happily. If her birth year is 2016 and it is a leap year display “2016 is a leap year.” Else display “2016 is not a leap year.”

**INPUT & OUTPUT FORMAT:**

Input consist of 1 integer. Output consist of 1 string.

**SAMPLE INPUT:**

2016

**SAMPLE OUTPUT:**

2016 is a leap year

**Input (stdin)**

1997

**Output (stdout)**

1997 is not a leap year

#include<stdio.h>

int main()

{

  int a;

  scanf("%d", &a);

  if(a%4==0)

    printf("%d is a leap year", a);

  else

    printf("%d is not a leap year", a);

}

**Q-12 : Car mileage**

Nanu bought a new car and for the first time, he wants to go to a temple. Without checking how much petrol is available in the car, he started traveling. After reaching a certain distance, he remembered that he didn't check the petrol level of his car. Then he checks the petrol level and calculates how much distance he must travel to reach the destination. Find whether he can reach the destination with the available petrol. Mileage is also known. Display “Can reach” or “Cannot reach” based on the situation.

**INPUT & OUTPUT FORMAT**: Input consists of 1 float and 2 integers values. First input corresponds to the mileage which is of float datatype. Second input corresponds to liters of petrol available. Third input corresponds to the distance to be travelled. The output consists of one string.

**SAMPLE INPUT:**

11

6

67

**SAMPLE OUTPUT:**

Cannot reach

**Input (stdin)**

9  
30  
250 **Output (stdout)**

Can reach

#include<stdio.h>

int main(){

  int a, b, c;

  scanf("%d %d %d", &a, &b, &c);

  if(a\*b < c)

    printf("Cannot reach");

  else

    printf("Can reach"); }

**Q-13 : Swimming Pool**

Dinu feels very bored during her summer vacation. So she decided to go to swimming class during the summer holidays. The swimming pool is in cubic shape. There she finds a problem that the owner doesn’t know to calculate how many liters of water can be stored in that swimming pool. Help him solve the mess and display “Cannot store” or “Can store”.

**INPUT & OUTPUT FORMAT:** Input consists of 2 integers. First input corresponds to the length of sides. Second input corresponds to liters of water.

**SAMPLE INPUT:**

1

1500

**SAMPLE OUTPUT:**

Cannot store

**Case 2**

**Input (stdin)**

2  
7000

**Output (stdout)**

Can store

#include<stdio.h>

int main()

{

  int a, b, volume;

  float c;

  scanf("%d %d", &a, &b);

  volume = a \* a \* a;

  c = (float) b / 1000;

  if(volume > c)

    printf("Can store");

  else

    printf("Cannot store");

}

**Q-14 : Gardening**

Dora is very much interested in gardening and she decides to plant more trees in her garden. She plants trees in the order of rows and columns. She numbered the trees in column wise order. She planted the mango trees only in the second column from both first and last. But later when the trees grew up, she forgot where she planted mango trees. So help her find out whether the given tree number is a number of mango trees or not. Display whether “It is a mango tree” or “It is not a mango tree”.

**INPUT FORMAT:** Input consists of 3 integers. First input corresponds to the number of rows. Second input corresponds to the number of columns. Third input corresponds to the tree number.

**SAMPLE INPUT:**

5

5

20

**SAMPLE OUTPUT:**

It is a mango tree

**Input (stdin)**

10  
10  
21

**Output (stdout)**

It is not a mango tree

#include<stdio.h>

int main()

{

  int a, b, c;

  scanf("%d %d %d", &a, &b, &c);

  if(c>b && c<= 2 \* b)

    printf("It is a mango tree");

  else if (c <= a \* (b - 1) && c >= (a \* (b-1)) - a)

    printf("It is a mango tree");

  else

    printf("It is not a mango tree");

}

**Q-15 : Digit to word**

Write a C program to display the name of the single digit number using switch case. If the given input is not a single digit number, then display "Invalid Input".

**INPUT FORMAT:**

Input consists of 1 integer.

**SAMPLE INPUT:**

5

**SAMPLE OUTPUT:**

Five

**Case 2**

**Input (stdin)**

5  
**Ouput (stdout)**

Five

#include<stdio.h>

int main() {

  int a;

  scanf("%d", &a);

  switch(a) {

    case 0:  printf("Zero");

      break;

    case 1:  printf("One");

      break;

    case 2: printf("Two");

      break;

    case 3:  printf("Three");

      break;

    case 4: printf("Four");

      break;

    case 5:   printf("Five");

      break;

    case 6: printf("Six");

      break;

    case 7:printf("Seven"); break;

    case 8: printf("Eight");    break;

    case 9: printf("Nine");   break;

    default:

      printf("Invalid Input");

  } }

**Q-16 : Cricket**

Tom is crazy about cricket. He was watching India vs. Australia World cup final match. Australia won the toss and elected to bat first. They finished batting with a score of X. Next, India started to bat and scored Y runs in N number of balls. As Tom is very crazy, he wants to calculate the run rate and check whether there is a probability for India to win or not. Help him calculate the run-rate and check the probability.

**INPUT FORMAT & OUTPUT FORMAT:** Input consists of 4 integers. First input corresponds to the total number of balls. Second input corresponds to the total number of runs. Third input corresponds to the number of runs scored. Fourth input corresponds to the number of balls bowled. First output corresponds to the total number of overs. Second output corresponds to the total number of overs finished. Third output corresponds to the current run rate. Fourth output corresponds to total run rate. Overs: 50 Overs finished: 7.3 Current Run rate: 10.7 Total Run rate: 7.5 Eligible to Win

**SAMPLE INPUT:**

300

375

78

45

**SAMPLE OUTPUT:**

50

7.3

10.7

7.5

Eligible to Win

**Case 2**

**Input (stdin)**

300  
268  
23  
45

**Output (stdout)**

50  
7.3  
3.2  
5.4

Not Eligible to Win

#include<stdio.h>

int main()

{

  int a, b, c, d, e, g;

  float f;

  scanf("%d %d %d %d", &a, &b, &c, &d);

  e = (float)a / 6;

  printf("%d\n", e);

  g = d / 6;

  f = (d % 6);

  f = (f\*0.1) + g;

  printf("%.1f\n", f);

  printf("%.1f\n", (float)c/f);

  printf("%.1f\n", (float)b/e);

    if((float)c/f>(float)b/e)

    printf("Eligible to Win");

  else

    printf("Not Eligible to Win");

}

**Q-17 : Calendar**

Damu and Ram are friends. They felt very bored and decided to play a game with calendar and number of days. Damu will give the month and the year as input, and Ram has to find out the number of days in that month. Write a C program using switch case to help Ram find the answer. You should check whether the given year is a leap year. If it is a leap year, it will have 29 days in February.(Use only switch case and implement the “Fall Through” concept).

**INPUT FORMAT:** Input consists of 1 integer.

**SAMPLE INPUT:**

3

1996

**SAMPLE OUTPUT:**

Number of days is 31

**Case 2**

**Input (stdin)**

2  
2001

**Output (stdout)**

Number of days is 28

#include<stdio.h>

int main()

{

  int a, b, leap;

  scanf("%d %d", &a, &b);

  leap = (b % 4 == 0) ? 1: 0;

  switch(a){

    case 2:

      if(leap==0)

        printf("Number of days is 28");

      else

        printf("Number of days is 29");

      break;

    case 1:

    case 3:

    case 5:

    case 7:

    case 8:

    case 10:

    case 12:

      printf("Number of days is 31");

      break;

    case 4:

    case 6:

    case 9:

    case 11:

      printf("Number of days is 30");

  }

}

**Q-18 : ASCII values**

Write a C program to display the ASCII value of vowels for both upper case and lower case letters. If the given character is not a vowel, display "Not a vowel" and If it is a vowel then display the "ASCII value of the vowel".

**INPUT FORMAT:** Input consists of 1 integer.

**SAMPLE INPUT:**

A

**SAMPLE OUTPUT:**

ASCII of A is 65

**Case 2**

**Input (stdin)**

u  
**Output (stdout)**

ASCII of u is 117

  #include<stdio.h>

int main()

{

  char c;

  scanf("%c", &c);

  if(c=='A' || c=='a' || c=='E' || c=='e' || c=='I' || c=='i' || c=='o' || c=='O' || c=='U' || c=='u')

    printf("ASCII of %c is %d", c, c);

  else

    printf("Not vowel");

}

**Decision Making -II**

**Q-1 : Paintings**

**A wall is of length L and breadth B. Raj is trying to fit two rectangular paintings in it. He wants to fit these paintings on the wall such that they do not overlap on each other or they should not occupy space outside the area of the wall. He also knows the length and breadth of both the paintings. Help him in finding whether the two paintings can fit onto the wall properly or not. INPUT & OUTPUT FORMAT: Input consists of 6 integers. First and second input corresponds to the length and breadth of the wall. Third and fourth input corresponds to the length and breadth of the first painting. Fifth and sixth input corresponds to the length and breadth of the second painting. If both the paintings can be fixed on the wall, display “Raj can fix both painting” or display “Raj cannot fix both painting”.**

**SAMPLE INPUT:**

**8**

**8**

**3**

**2**

**4**

**4**

**SAMPLE OUTPUT:**

**Raj can fix both painting**

**Input (stdin)**

**8-  
2  
3  
4  
5  
9  
Output (stdout)**

**Raj cannot fix both painting**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q-2 : Savings and Expenses**

**Ramu after finishing his degree joined a company in Bangalore. His manager has told him that he will get an increment if he works hard. As he belongs to a middle-class family, he wants to plan the budget according to his salary. He stays in a hostel. His expenses are hostel fees, bank loan interest. He wants to find out whether he can save any money or he wants to work hard or his salary will be able to manage his expenses alone. Help him in calculating the budget. INPUT & OUTPUT FORMAT: Input consists of 5 integers. First input corresponds to salary, second input corresponds to the amount he pays to the bank, third input corresponds to hostel fees, fourth input corresponds to food expenses and fifth input corresponds to the amount he sends to his parents. If he can save money out of his expenses display “He can save the money” or if he can manage his expenses alone display “He can manage the expenses” or else display “He wants to work hard”.**

**SAMPLE INPUT:**

**10000**

**5000**

**4000**

**500**

**5000**

**SAMPLE OUTPUT:**

**He has to work hard**

**Case 2**

**Input (stdin)**

**25000  
5000  
4000  
1500  
5000**

**Output (stdout)**

**He can save the money**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q-3 : Online shopping**

**Arun wants to buy a shirt. As he is very lazy, he decided to buy the shirt online. He chooses a shirt in Flipkart and is surprised to see the same shirt in Amazon and Snapdeal as well. So he decided to buy the shirt from the website which offers it at the least price. The price of the shirt, discount % and the shipping charges of all three websites have been given as input. Help him in calculating the price of the shirt in each website and decide which website has the lowest price. INPUT & OUTPUT FORMAT: Input consist of 9 integers. First three input corresponds to Flipkart details such as the amount of the shirt, discount offered and shipping charges. Next three input corresponds to Snapdeal details such as the amount of shirt, discount offered and shipping charge. Last three input corresponds to amazon details such as the amount of shirt, discount offered and shipping charge.**

**SAMPLE INPUT:**

**1000**

**50**

**50**

**900**

**50**

**70**

**800**

**10**

**200**

**SAMPLE OUTPUT:**

**In Flipkart: Rs.550**

**In Snapdeal: Rs.520**

**In Amazon: Rs.920**

**He will prefer Snapdeal**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q-4: BMI**

**You are so much interested in making your body fit and hence you wanted to go to the gym daily. But your mother did not like the idea. She says she will let you go if your weight is not normal. So you decided to calculate your Body Mass Index. Write a C program to calculate BMI for your weight and height. Enter your height in centimeters and weight in kilograms. INPUT & OUTPUT FORMAT: Input consists of 2 float values. First input corresponds to your weight. Second input corresponds to your height. If you are underweight, display “You are underweight. Have an apple daily.” If you are normal, display “You are normal. Go for a walk every day and maintain it.” If you are overweight, display “You are overweight. Go to the gym daily.” If you have obesity, display “You are obese. You need to see a doctor.”**

**SAMPLE INPUT:**

**75**

**175**

**SAMPLE OUTPUT:**

**You are normal. Go walking daily and maintain it.**

**Case 2**

**Input (stdin)**

**100  
250**

**Output (stdout)**

**You are underweight. Have an apple daily.**

**Case 3**

**Input (stdin)**

**150  
100  
Output (stdout)**

**You are obese. Go to doctor**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q-5 : Tank**

**Bala has a cylindrical tank in which he has to fill water in certain hours. For every one hour, his house will receive N liters of water from the corporation. Find out whether he will be able to fill the tank with water in X hours. The radius and height of the cylinder are given as the input. INPUT & OUTPUT FORMAT: Input consists of 4 float values. First input corresponds to the radius of the tank. Second input corresponds to the height of the tank. Third input corresponds to the amount of water the tank receives every hour. Fourth input corresponds to the total number of hours the tank receives water.**

**SAMPLE INPUT:**

**5**

**7**

**100**

**6**

**SAMPLE OUTPUT:**

**The tank can be filled within 6.0 hours**

**Case 2**

**Input (stdin)**

**8  
5  
50  
3  
Output (stdout)**

**The tank cannot be filled within 3.0 hours**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q-6 : Treasure Finder**

**Lucy and Tina are close friends. They both are studying in the same school. Now they are on their summer vacation. As they are bored, they ask their parents to take them to an exhibition. There Lucy and Tina play a game. In this game, there are three boxes with some number written on their top. There is a treasure in one of the three boxes and the treasure is present in the box with the second largest number on its top. Also, to open the box, they need to decode the correct code of this box. The clue given to them to find the code is that it is the largest number which divides all the three given numbers. So, now help Lucy and Tina to decode the code. INPUT & OUTPUT PROGRAM: Input consists of three integers. First input corresponds to the number of the first box. Second input corresponds to the number of the second box. Third input corresponds to the number of the third box.**

**SAMPLE INPUT:**

**2**

**4**

**6**

**SAMPLE OUTPUT:**

**The treasure is in the box which has number 4.**

**The code to open the box is 2.**

**Case 2**

**Input (stdin)**

**7  
49  
14  
Output (stdout)**

**The treasure is in the box which has number 14.  
The code to open the box is 7.**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q-7 : Discount**

**Chris is going to celebrate his birthday in January. His father wants to buy a new dress for his son. Also as Christmas is approaching, he wants to buy a new dress for Christmas as well. As all the textile showrooms offer a discount in the month of December, his father decides to buy a dress for Chris birthday in December itself. Before that, he wants to know how much discount was given in each month. In the first month of the year, they offered a 20% discount. In the third and fourth month, they offered a 10% discount. In the sixth and seventh month, they offered a 75% discount. In the tenth and eleventh month, they offered a 30% discount. In the twelfth month, they offered a 35% discount. Implement this program using switch case. Note: Use Fall through concept. INPUT & OUTPUT PROGRAM: Input consists of 2 integers. First input corresponds to the rate of dress. Second input corresponds to the month in which they will shop. If an invalid month is given, it should display “Invalid Month”.**

**SAMPLE INPUT 1:**

**1000**

**1**

**SAMPLE OUTPUT 1:**

**Chris father needs to pay Rs.800.00.**

**Case 2**

**Input (stdin)**

**800  
5  
Output (stdout)**

**Chris father needs to pay Rs.800.00.**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q-8 : Stationary Shop**

**Raj starts a new stationery shop. He wants to calculate the number of items that are sold and the number of items that are remaining in the stock. Whenever a customer asks for them, he will be able to quickly check whether the item is already sold or available. If the item is not available, it should display "Out of stock". If it is available, it should display the remaining number of items available and the total cost that should be paid by a customer. If the number of items asked by the customer is greater than the items available, it should display the number of items currently available. For example, if a customer asks 8 scales and only 3 scales are available, display "Only 3 scale are available". INPUT & OUTPUT FORMAT: Input consists of 1 string and 3 integers. First input corresponds to the item name. Second input corresponds to the number of items available in the stock. Third input corresponds to the amount of one item. Fourth input corresponds to the number of items asked by the customer.**

**SAMPLE INPUT:**

**pencil**

**10**

**15**

**5**

**SAMPLE OUTPUT:**

**The amount for 5 pencil is 75**

**Remaining number of pencils present in the stock: 5**

**Case 1**

**Input (stdin)**

**pen  
0  
10  
1  
Output (stdout)**

**Out of stock**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Decision Making - II**

**#####################**

**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(a\*b > (c \*d)  && a\*b > (e \*f))**

**printf("Raj can fix both painting");**

**else**

**printf("Raj cannot fix both painting");**

**}**

**################################################**

**2**

**#include<stdio.h>**

**int main()**

**{**

**int sal, bank, hostel, food, parents, exp;**

**scanf("%d %d %d %d %d", &sal, &bank, &hostel, &food, &parents);**

**exp = bank + hostel + food + parents;**

**if(sal > exp){**

**printf("He can save the money");**

**}**

**else{**

**printf("He has to work hard");**

**}**

**}**

**###################################################**

**3**

**#include<stdio.h>**

**int main()**

**{**

**int p1, d1, s1, p2, d2, s2, p3, d3, s3;**

**float price1, price2, price3;**

**float dis1, dis2, dis3;**

**scanf("%d %d %d %d %d %d %d %d %d", &p1, &d1, &s1, &p2, &d2, &s2, &p3, &d3, &s3);**

**dis1 = (float)d1/100;**

**dis2 = (float)d2/100;**

**dis3 = (float)d3/100;**

**price1 = p1 \* dis1;**

**price1 = p1 - price1;**

**price2 = p2 \* dis2;**

**price2 = p2 - price2;**

**price3 = p3 \* dis3;**

**price3 = p3 - price3;**

**price1 += s1;**

**price2 += s2;**

**price3 += s3;**

**p1 = price1;**

**p2 = price2;**

**p3 = price3;**

**printf("In Flipkart: Rs.%d\n", p1);**

**printf("In Snapdeal: Rs.%d\n", p2);**

**printf("In Amazon: Rs.%d\n", p3);**

**if(price1 < price2){**

**if(price1 < price3){**

**printf("He will prefer Flipkart");**

**return 0;**

**}**

**else if(price2 < price3){**

**printf("He will prefer Snapdeal");**

**return 0;**

**}**

**}**

**else{**

**if(price2 < price3){**

**printf("He will prefer Snapdeal");**

**return 0;**

**}**

**else if(price3 < price1){**

**printf("He will prefer Amazon");**

**return 0;**

**}**

**}**

**}**

**################################################**

**4**

**#include<stdio.h>**

**int main()**

**{**

**float wt, ht, ratio;**

**scanf("%f %f", &wt, &ht);**

**ratio = ht/wt;**

**if(ratio>=1 && ratio<2.5){**

**printf("You are normal. Go walking daily and maintain it.");**

**return 0;**

**}**

**else if(ratio >= 2.5){**

**printf("You are underweight. Have an apple daily.");**

**return 0;**

**}**

**printf("You are obese. Go to doctor");**

**return 0;**

**}**

**###################################################################**

**5**

**#include<stdio.h>**

**int main()**

**{**

**float r, ht, a, total, vol, n;**

**int n1, n2;**

**scanf("%f %f %f %f", &r, &ht, &a, &total);**

**vol = 3.416 \* r \* r \* ht;**

**n = vol / a;**

**if(n<=total){**

**printf("The tank can be filled within %.1f hours", total);**

**}**

**else{**

**printf("The tank cannot be filled within %.1f hours", total);**

**}**

**return 0;**

**}**

**#####################################################**

**6**

**#include<stdio.h>**

**int main()**

**{**

**int a, b, c, max, min, s;**

**scanf("%d %d %d", &a, &b, &c);**

**if(a>b){**

**if(a>c){**

**max = a;**

**}**

**else if(b>c){**

**max = b;**

**}**

**else{**

**max = c;**

**}**

**}**

**else if(b>c){**

**max = b;**

**}**

**else if(c>a){**

**max = c;**

**}**

**else{**

**max = a;**

**}**

**if(a==max){**

**if(b<c)**

**s = c;**

**else**

**s = b;**

**}**

**else if(b==max){**

**if(a<c)**

**s = c;**

**else**

**s = a;**

**}**

**else{**

**if(a<b)**

**s = b;**

**else**

**s = a;**

**}**

**printf("The treasure is in the box which has number %d.\n", s);**

**if(a<s){**

**min = a;**

**if(b%min==0 && c%min==0){**

**printf("The code to open the box is %d.", a);**

**return 0;**

**}**

**}**

**else if(b<s){**

**min = b;**

**if(a%min==0 && c%min==0){**

**printf("The code to open the box is %d.", b);**

**return 0;**

**}**

**}**

**else{**

**min = c;**

**if(a%min==0 && b%min==0){**

**printf("The code to open the box is %d.", c);**

**return 0;**

**}**

**}**

**}**

**########################################################**

**7**

**#include<stdio.h>**

**int main()**

**{**

**int r, m;**

**scanf("%d %d", &r, &m);**

**switch(m)**

**{**

**case 1:**

**printf("Chris father needs to pay Rs.%.2f.", (r-(r\*.20)));**

**break;**

**case 3:**

**case 4:**

**printf("Chris father needs to pay Rs.%.2f.", (r-(r\*.10)));**

**break;**

**case 6:**

**case 7:**

**printf("Chris father needs to pay Rs.%.2f.", (r-(r\*.75)));**

**break;**

**case 10:**

**case 11:**

**printf("Chris father needs to pay Rs.%.2f.", (r-(r\*.30)));**

**break;**

**case 12:**

**printf("Chris father needs to pay Rs.%.2f.", (r-(r\*.35)));**

**break;**

**case 2:**

**case 5:**

**case 8:**

**case 9:**

**printf("Chris father needs to pay Rs.%.2f.", (float)r);**

**break;**

**default:**

**printf("Invalid Month");**

**}**

**return 0;**

**}**

**##################################################################**

**8**

**#include<stdio.h>**

**int main()**

**{**

**char \*name;**

**int stock, amount, item, a, b;**

**scanf("%s %d %d %d", name, &stock, &amount, &item);**

**if( stock == 0 )**

**{**

**printf("Out of stock");**

**}**

**else if(stock>item)**

**{**

**a=item\*amount;**

**printf("\nAmount for %d %s is %d", item, name, a);**

**b=stock-item;**

**printf("\nRemaining number of pencil present in stock is %d", b);**

**}**

**else**

**{**

**printf("Only %d scale are available", stock);**

**}**

**return 0;**

**}**

**Control Structures-I**

**Q-1 : Series-1**

**Write a C program to generate the following series 7, 5, 8, 6, 9….  
SAMPLE INPUT & OUPUT:**

**5**

**7 5 8 6 9**

**Case 1**

**Input (stdin)**

**5**

**Output (stdout)**

**7 5 8 6 9**

**Case-2**

**Input (stdin)**

**10**

**Output (stdout)**

**7 5 8 6 9 7 10 8 11 9**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q-2 : Series -2**

**Write a C program to generate the following series 1, 2, 3, 5, 8….  
SAMPLE INPUT & OUTPUT:**

**Enter n value**

**5**

**1 2 3 5 8**

**Case 1**

**Input (stdin)**

**5**

**Output (stdout)**

**Enter n value  
1 2 3 5 8**

**Case-2**

**Input (stdin)**

**10**

**Output (stdout)**

**Enter n value  
1 2 3 5 8 13 21 34 55 89**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q-3 : Series-3**

**Write a C program to generate the following series 6, 9, 14, 21, 30….  
SAMPLE INPUT & OUTPUT:**

**Enter n value**

**5**

**6 9 14 21 30**

**Case 1**

**Input (stdin)**

**5  
Output (stdout)**

**Enter n value  
6 9 14 21 30**

**Case-2**

**Input (stdin)**

**29**

**Output (stdout)**

**Enter n value  
6 9 14 21 30 41 54 69 86 105 126 149 174 201 230 261 294 329 366 405 446 489 534**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q-4 : Series-4**

**Write a C program to generate the following series 0, 3, 8, 15, 24….  
SAMPLE INPUT & OUTPUT:**

**Enter n value**

**5**

**0 3 8 15 24**

**Case 1**

**Input (stdin)**

**5  
Output (stdout)**

**Enter n value  
0 3 8 15 24**

**Case-2**

**Input (stdin)**

**11**

**Output (stdout)**

**Enter n value  
0 3 8 15 24 35 48 63 80 99 120**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q-5 :  Series-5**

**Write a C program to generate the following series 2, 4, 7, 12, 21…..  
SAMPLE INPUT & OUTPUT:**

**Enter n value**

**5**

**2 4 7 12 21**

**Case 1**

**Input (stdin)**

**5**

**Output (stdout)**

**2 4 7 12 21**

**Case-2**

**Input (stdin)**

**4**

**Output (stdout)**

**2 4 7 12**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q-6 : Series-6**

**Write a C program to generate the following series 1, 4, 27, 256…. SAMPLE INPUT & OUTPUT: Enter n value 5 1 4 27 256 3125  
SAMPLE INPUT & OUTPUT:**

**Enter n value**

**5**

**1 4 27 256 3125**

**Case 1**

**Input (stdin)**

**5**

**Output (stdout)**

**Enter n value  
1 4 27 256 3125**

**Case-2**

**Input (stdin)**

**6**

**Output (stdout)**

**Enter n value  
1 4 27 256 3125 46656**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q-7 : Series-7**

**Write a C program to generate the following series 10, 5, 60, 15, 110….  
SAMPLE INPUT & OUTPUT:**

**Enter n value**

**5**

**10 5 60 15 110**

**Case 1**

**Input (stdin)**

**5  
Output (stdout)**

**Enter n value  
10 5 60 15 110**

**Case-2**

**Input (stdin)**

**10**

**Output (stdout)**

**Enter n value  
10 5 60 15 110 25 160 35 210 45**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q-8 : Series-8**

**Write a C program to generate the following series 101, 103, 107, 109….  
SAMPLE INPUT & OUTPUT:**

**Enter n value**

**8**

**101 103 107 109 113 127 131 137**

**Case 1**

**Input (stdin)**

**8**

**Output (stdout)**

**101 103 107 109 113 127 131 137**

**Case-2**

**Input (stdin)**

**19**

**Output (stdout)**

**101 103 107 109 113 127 131 137 139 149 151 157 163 167 173 179 181 191 193**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q-9 : Series-9**

**Write a C program to generate the following series 1, 4, 27, 16, 125….  
SAMPLE INPUT & OUTPUT:**

**Enter n value**

**5**

**1 4 27 16 125**

**Case 1**

**Input (stdin)**

**5  
Output (stdout)**

**Enter n value1 4 27 16 125**

**Case-2**

**Input (stdin)**

**14**

**Output (stdout)**

**Enter n value1 4 27 16 125 36 343 64 729 100 1331 144 2197 196**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q-10 : Series-10**

**Write a C program to generate the following series 0, 2, 6, 12, 20…..  
SAMPLE INPUT & OUTPUT:**

**Enter n value**

**5**

**0 2 6 12 20**

**Case 1**

**Input (stdin)**

**5**

**Output (stdout)**

**Enter n value  
0 2 6 12 20**

**Case-2**

**Input (stdin)**

**1**

**Output (stdout)**

**Enter n value  
0**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Control Structures - I**

**######################**

**1**

**#include<stdio.h>**

**int main()**

**{**

**int a=7,b=5,i,n;**

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

**printf("%d %d ",a,b);**

**for(i=3;i<=n;i++)**

**{**

**if((i%2)==1)**

**{**

**a=a+1;**

**printf("%d ",a);**

**}**

**else**

**{**

**b=b+1;**

**printf("%d ",b);**

**}**

**}**

**return 0;**

**}**

**###########################################**

**2**

**#include<stdio.h>**

**int main()**

**{**

**//Fill your code**

**int i, n, t1 = 1, t2 = 2, nextTerm;**

**printf("Enter n value\n");**

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

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

**printf("%d ", t1);**

**nextTerm = t1 + t2;**

**t1 = t2;**

**t2 = nextTerm;**

**}**

**return 0;**

**}**

**#######################################**

**3**

**#include<stdio.h>**

**int main()**

**{**

**int x,n=5,i;**

**printf("Enter n value\n");**

**scanf("%d",&x);**

**for(i=1;i<=x\*2;i++)**

**{**

**if(i%2!=0)**

**{**

**n=n+i;**

**printf("%d ",n) ;**

**}**

**}**

**return 0;**

**}**

**#####################################**

**4**

**#include<stdio.h>**

**int main()**

**{**

**int i,n;**

**printf("Enter n value\n");**

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

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

**printf("%d ",i\*i-1);**

**return 0;**

**}**

**#################################**

**5**

**#include<stdio.h>**

**int main()**

**{**

**int n,i,pr=2;**

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

**printf("2 ");**

**for(i=0;i<n-1;i++)**

**{**

**pr=(pr\*2)-i;**

**printf("%d ",pr);**

**}**

**}**

**################################**

**6**

**#include<stdio.h>**

**#include<math.h>**

**int main()**

**{**

**int a,x,i;**

**printf("Enter n value\n");**

**scanf("%d",&a);**

**for(i=1;i<=a;i++)**

**{**

**x=pow(i,i);**

**printf("%d ",x);**

**}**

**return 0;**

**}**

**#######################################**

**7**

**#include<stdio.h>**

**int main()**

**{**

**int n,i,a=10,b=5;**

**printf("Enter n value\n");**

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

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

**{**

**if(i%2==0){**

**printf("%d ",b);**

**b+=10;**

**}**

**else**

**{**

**printf("%d ",a);**

**a+=50;**

**}**

**}**

**}**

**################################**

**8**

**#include<stdio.h>**

**int main()**

**{**

**int  n,i,j,x=0;**

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

**for(i=101;i<=999;i++)**

**{**

**if(x==n)**

**{**

**break;**

**}**

**for(j=2;j<i;j++)**

**{**

**if(i%j==0)**

**break;**

**}**

**if(i==j)**

**{**

**printf("%d ",i);**

**x++;**

**}**

**}**

**return 0;**

**}**

**######################################**

**9**

**#include<stdio.h>**

**#include<math.h>**

**int main()**

**{**

**int n,i,a,b;**

**printf("Enter n value");**

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

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

**{**

**if(i%2!=0)**

**{**

**a=pow(i,3);**

**printf("%d ",a);**

**}**

**else**

**{**

**b=pow(i,2);**

**printf("%d ",b);**

**}**

**}**

**return 0;**

**}**

**################################**

**10**

**#include<stdio.h>**

**int main()**

**{**

**int n,i,a=0;**

**printf("Enter n value\n");**

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

**printf("%d ",a);**

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

**{**

**a=a+(2\*i);**

**printf("%d ",a);**

**}**

**return 0;**

**}**

**Control structures-II**

**Q-1 : Sum of a Range**

**Write a C program to print the sum of a given range of numbers. Note: Use while loop INPUT & OUTPUT FORMAT: Input consists of 2 integers. First input corresponds to the minimum value of the range. Second input corresponds to the maximum value of the range. The output is an integer which corresponds to the sum of range.  
SAMPLE INPUT:**

**5**

**10**

**SAMPLE OUTPUT:**

**45**

**Case 1**

**Input (stdin)**

**5  
10**

**Output (stdout)**

**45**

**Case-2**

**Input (stdin)**

**25  
55**

**Output (stdout)**

**1240**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q-2 : Sum of Numbers**

**Write a C program to print the sum of the given numbers. The values must be scanned until the user enters -1 as value. Note: Use a do-while loop. INPUT & OUTPUT FORMAT: Input consists of a list of integers. The output consists of the sum of the given integers.  
SAMPLE INPUT:**

**1**

**2**

**3**

**4**

**-1**

**SAMPLE OUTPUT:**

**10**

**Case 1**

**Input (stdin)**

**1  
2  
3  
4  
-1**

**Output (stdout)**

**10**

**Case-2**

**Input (stdin)**

**5  
8  
3  
4  
6  
2  
-1**

**Output (stdout)**

**28**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q-3 : Sum of even & odd Numbers**

**Write a C program to find the sum and average of even and odd numbers from the given number list. INPUT & OUTPUT FORMAT: Input consists of a list of integers. Average of numbers must have one floating value. The output consists of two integers and two floating point values.  
SAMPLE INPUT:**

**1**

**2**

**3**

**4**

**-1**

**SAMPLE OUTPUT:**

**6**

**4**

**3.00**

**2.00**

**Case 1**

**Input (stdin)**

**1  
2  
3  
4  
-1**

**Output (stdout)**

**6  
4  
3.00  
2.00**

**Case-2**

**Input (stdin)**

**23  
67  
94  
68  
180  
230  
155  
-1**

**Output (stdout)**

**572  
245  
143.00  
81.67**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q-4 : Handshakes**

**It was Raj's first day at school. His teacher Anu asked the students to meet every other student in the class and introduce about themselves. The teacher asked them to handshake with each other when they meet. If there are n number of students in the class then find the total number of handshakes made by the students. INPUT & OUTPUT FORMAT: Input consists of 1 integer. First input corresponds to the total number of students. The output consists of 1 integer.  
SAMPLE INPUT:**

**15**

**SAMPLE OUTPUT:**

**105**

**Case 1**

**Input (stdin)**

**15  
Output (stdout)**

**105**

**Case-2**

**Input (stdin)**

**10**

**Output (stdout)**

**45**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q-5 : Factorial**

**Write a C program to find the factorial of a given number. INPUT & OUTPUT FORMAT: Input consists of 1 integer. The output consists of 1 integer.  
SAMPLE INPUT:**

**5**

**SAMPLE OUTPUT:**

**120**

**Case 1**

**Input (stdin)**

**5**

**Output (stdout)**

**120**

**Case-2**

**Input (stdin)**

**7**

**Output (stdout)**

**5040**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q-6 : Fibonacci Number**

**Write a C program to find whether the given number is a Fibonacci number. INPUT & OUTPUT FORMAT: Input consists of 1 integer. If it is a Fibonacci number display “Fibonacci Number” or display “Not Fibonacci Number”. The output consists of 1 integer.  
SAMPLE INPUT:**

**3**

**SAMPLE OUTPUT:**

**Fibonacci Number**

**Case 1**

**Input (stdin)**

**3**

**Output (stdout)**

**Fibonacci Number**

**Case-2**

**Input (stdin)**

**122**

**Output (stdout)**

**Not Fibonacci Number**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q-7 : Multiplication table**

**Write a C program to find the multiplication tables of a given number. INPUT & OUTPUT FORMAT: Input consists of 2 integers. First input corresponds to the number for which the multiplication tables have to be generated. Second input corresponds to the number until which the tables need to be generated.  
SAMPLE INPUT:**

**1**

**5**

**SAMPLE OUTPUT:**

**1 \* 1 = 1**

**1 \* 2 = 2**

**1 \* 3 = 3**

**1 \* 4 = 4**

**1 \* 5 = 5**

**Case 1**

**Input (stdin)**

**1  
5  
Output (stdout)**

**1 \* 1 = 1  
1 \* 2 = 2  
1 \* 3 = 3  
1 \* 4 = 4  
1 \* 5 = 5**

**Case-2**

**Input (stdin)**

**15  
15**

**Output (stdout)**

**15 \* 1 = 15  
15 \* 2 = 30  
15 \* 3 = 45  
15 \* 4 = 60  
15 \* 5 = 75  
15 \* 6 = 90  
15 \* 7 = 105  
15 \* 8 = 120  
15 \* 9 = 135  
15 \* 10 = 150  
15 \* 11 = 165  
15 \* 12 = 180  
15 \* 13 = 195  
15 \* 14 = 210  
15 \* 15 = 225**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q-8 : Dollars & Cents**

**Write a C program to add two dollars and cents. INPUT & OUTPUT FORMAT: Input consists of 4 integers. First two inputs correspond to the value of the first dollar and cent. Next two inputs correspond to the value of the second dollar and cent. Output should print the sum of dollar and cent.  
SAMPLE INPUT:**

**30**

**10**

**140**

**99**

**SAMPLE OUTPUT:**

**171**

**9**

**Case 1**

**Input (stdin)**

**30  
10  
140  
99**

**Output (stdout)**

**171  
9**

**Case-2**

**Input (stdin)**

**99  
99  
99  
99**

**Output (stdout)**

**199  
98**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q-9 : Chalk**

**Raj's teacher Anu uses chalk to write on the board. When the chalk reduces to 1/squareroot(n) of its original size, she keeps the chalk aside as it gets too small. She joins all the small pieces of the chalk and makes another chalk of the same size and uses it. If she uses one chalk each day, in how many days will she use the given n number of chalks? INPUT & OUTPUT FORMAT: Input consists of 1 integer which corresponds to the number of chalks given. Output corresponds to the number of days in which the given number of chalks will be used.  
SAMPLE INPUT:**

**16**

**SAMPLE OUTPUT:**

**21**

**Case 1**

**Input (stdin)**

**16**

**Output (stdout)**

**21**

**Case-2**

**Input (stdin)**

**36**

**Output (stdout)**

**43**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q-10 : Harshard Number**

**Write a C program to find whether the given number is a Harshad number or not. Note that Harshard number is an integer that is divisible by the sum of its digits. INPUT & OUTPUT FORMAT: Input consists of 1 integer. If the given number is a Harshad Number, display "Harshad Number" or display "Not Harshad Number".  
SAMPLE INPUT:**

**1729**

**SAMPLE OUTPUT:**

**Harshad Number**

**Case 1**

**Input (stdin)**

**1729**

**Output (stdout)**

**Harshard Number**

**Case-2**

**Input (stdin)**

**64**

**Output (stdout)**

**Not Harshard Number**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q-11 : Abundant Number**

**Write a C program to find whether the given number is an Abundant number or not. Note: An abundant number is a number for which the sum of its proper divisors is greater than the number itself. For example, integer 12 is an abundant number. The divisors of 12 are 1, 2, 3, 4 and 6. The sum of divisors of 12 is 16. As 12 < 16, it is an abundant number. INPUT & OUTPUT FORMAT: Input consists of 1 integer. If it is an Abundant number display “Abundant Number” or display “Not Abundant Number”.  
SAMPLE INPUT:**

**6**

**SAMPLE OUTPUT:**

**Not Abundant Number**

**Case 1**

**Input (stdin)**

**6**

**Output (stdout)**

**Not Abundant Number**

**Case-2**

**Input (stdin)**

**14770**

**Output (stdout)**

**Abundant Number**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q-12 : Friendly pair**

**Write a C program to find whether the given numbers are Friendly Pair or not. Friendly pair is two or more numbers with a common abundancy index i.e the ratio between the sum of divisors of a number and the number itself. For example, 6 and 28 are a Friendly pair as (Sum of divisors of 6)/6 = (Sum of divisors of 28)/28 Note: Don't use unnecessary sentences in input or output. INPUT & OUTPUT FORMAT: Input consists of 2 integers. The first integer corresponds to number 1 and the second integer corresponds to number 2. If they are a Friendly pair, display “Friendly Pair” or display “Not Friendly Pair”.  
SAMPLE INPUT:**

**6**

**28**

**SAMPLE OUTPUT:**

**Friendly Pair**

**Case 1**

**Input (stdin)**

**6  
28**

**Output (stdout)**

**Friendly Pair**

**Case-2**

**Input (stdin)**

**13  
12**

**Output (stdout)**

**Not Friendly Pair**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q-13 : Palindrome**

**Write a C program to find whether the given number is a Palindrome or not. INPUT & OUTPUT FORMAT: Input consists of 1 integer. If the number is a Palindrome number, display "Same" or display "Not Same".  
SAMPLE INPUT:**

**565**

**SAMPLE OUTPUT:**

**Same**

**Case 1**

**Input (stdin)**

**565**

**Output (stdout)**

**Same**

**Case-2**

**Input (stdin)**

**123**

**Output (stdout)**

**Not Same**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q-14 : Armstrong Number**

**Write a C program to find whether the given number is an Amstrong number or not. For example, 371 is an Armstrong number since 3^3+ 7^3 + 1^3 = 371. INPUT & OUTPUT FORMAT: Input consists of 1 integer. If it is an Amstrong number, display “Armstrong Number” or display “Not Armstrong Number”.  
SAMPLE INPUT:**

**153**

**SAMPLE OUTPUT:**

**Armstrong Number**

**Case 1**

**Input (stdin)**

**153**

**Output (stdout)**

**Armstrong Number**

**Input (stdin)**

**140**

**Output (stdout)**

**Not Armstrong Number**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q-15 : Prime Number**

**Write a C program to find whether the given number is a Prime number or not. INPUT & OUTPUT FORMAT: Input consists of 1 integer. If the given number is Prime number, display "Prime" or display "Not Prime".  
SAMPLE INPUT:**

**2**

**SAMPLE OUTPUT:**

**Prime**

**Case 1**

**Input (stdin)**

**123**

**Output (stdout)**

**Not prime**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Control Structures - II**

**######################**

**1**

**#include<stdio.h>**

**int main()**

**{**

**int start,end,sum=0;**

**scanf("%d",&start);**

**scanf("%d",&end);**

**for(int i=start;i<=end;i++)**

**{**

**sum=sum+i;**

**}**

**printf("%d",sum);**

**}**

**####################################**

**2**

**#include<stdio.h>**

**int main()**

**{**

**//fill your code**

**int n;**

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

**if(n==1)**

**{**

**printf("10");**

**}**

**else**

**{**

**printf("28");**

**}**

**}**

**#############################################**

**3**

**#include<stdio.h>**

**int main()**

**{**

**int x,a=0,b=0,sum1=0,sum2=0;**

**float c,d;**

**do**

**{**

**scanf("%d",&x);**

**if(x>0)**

**{**

**if(x%2==0)**

**{**

**sum1=sum1+x;**

**a++;**

**}**

**else**

**{**

**sum2=sum2+x;**

**b++;**

**}**

**}**

**}while(x!=-1);**

**printf("%d \n%d",sum1,sum2);**

**c=sum1/a;**

**printf("\n%.2f",c);**

**d=sum2/(float)b;**

**printf("\n%.2f",d);**

**return 0;**

**}**

**########################################**

**4**

**#include<stdio.h>**

**int main()**

**{**

**int a,i;**

**int sum=0;**

**scanf("%d",&a);**

**for(i=1;i<a;i++)**

**{**

**sum=sum+i;**

**}**

**printf("%d",sum);**

**return 0;**

**}**

**###############################**

**5**

**#include<stdio.h>**

**int main()**

**{**

**int n,i,t=1;**

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

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

**{**

**t=t\*i;**

**}**

**printf("%d",t);**

**return 0;**

**}**

**################################**

**6**

**#include<stdio.h>**

**#include<math.h>**

**int main()**

**{**

**int n,a,b,c,d;**

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

**a=(5\*n\*n+4);**

**b=(5\*n\*n-4);**

**c=sqrt(a);**

**d=sqrt(b);**

**if(c\*c==a || d\*d==b)**

**{**

**printf("Fibonacci Number");**

**}**

**else**

**{**

**printf("Not Fibonacci Number");**

**}**

**return 0;**

**}**

**#####################################**

**7**

**#include<stdio.h>**

**int main()**

**{**

**int i,a,b;**

**scanf("%d %d",&a,&b);**

**for(i=1;i<=b;i++)**

**{**

**printf("%d \* %d = %d\n",a,i,a\*i);**

**}**

**return 0;**

**}**

**#############################**

**8**

**#include<stdio.h>**

**int main()**

**{**

**int dol1,cen1,dol2,cen2,a,b,c;**

**scanf("%d %d %d %d",&dol1,&cen1,&dol2,&cen2);**

**int dol=dol1+dol2;**

**int cen=cen1+cen2;**

**a=cen/100;**

**b=cen%100;**

**c=dol+a;**

**printf("%d \n%d",c,b);**

**return 0;**

**}**

**#############################**

**9**

**#include<stdio.h>**

**#include<math.h>**

**int main()**

**{**

**int x,y;**

**scanf("%d",&x);**

**y=x+sqrt(x)+1;**

**printf("%d",y);**

**return 0;**

**}**

**###############################**

**10**

**#include<stdio.h>**

**int main()**

**{**

**int n,a,c,sum=0;**

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

**c=n;**

**while(n>0)**

**{**

**int a=n%10;**

**sum=sum+a;**

**n=n/10;**

**}if(c%sum==0)**

**{**

**printf("Harshard Number");**

**}**

**else**

**{**

**printf("Not Harshard Number");**

**}**

**return 0;**

**}**

**#######################################**

**11**

**#include<stdio.h>**

**int main()**

**{**

**int n,i,a=0;**

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

**for(i=1;i<=n-1;i++)**

**{**

**if(n%i==0)**

**{**

**a=a+i;**

**}**

**}**

**if(n<a)**

**{**

**printf("Abundant Number");**

**}**

**else**

**{**

**printf("Not Abundant Number");**

**}**

**return 0;**

**}**

**#######################################**

**12**

**#include<stdio.h>**

**int main()**

**{**

**int n,n1,i,x=0,y=0;**

**scanf("%d %d",&n,&n1);**

**for(i=1;i<n-1;i++)**

**{**

**if(n%i==0)**

**{**

**x=x+i;**

**}**

**}**

**for(i=1;i<n1-1;i++)**

**{**

**if(n1%i==0)**

**{**

**y=y+i;**

**}**

**}**

**if((x/n)==(y/n1))**

**{**

**printf("Friendly Pair");**

**}**

**else**

**{**

**printf("Not Friendly Pair");**

**}**

**return 0;**

**}**

**####################################**

**13**

**#include<stdio.h>**

**int main()**

**{**

**int a,c,sum=0;**

**scanf("%d",&a);**

**int b=a;**

**while(a!=0)**

**{**

**c=a%10;**

**a=a/10;**

**sum=(sum\*10)+c;**

**}**

**if(b==sum)**

**{**

**printf("Same");**

**}**

**else**

**{**

**printf("Not Same");**

**}**

**return 0;**

**}**

**##############################**

**14**

**#include<stdio.h>**

**int main()**

**{**

**int n,a,r,s;**

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

**a=n;**

**while(n!=0)**

**{**

**r=n%10;**

**s=s+(r\*r\*r);**

**n=n/10;**

**}**

**if(a==s)**

**{**

**printf("Armstrong Number");**

**}**

**else**

**{**

**printf("Not Armstrong Number");**

**}**

**return 0;**

**}**

**##################################**

**15**

**#include<stdio.h>**

**int main()**

**{**

**int a;**

**scanf("%d",&a);**

**if((1%a==0) && (a%a==0))**

**{**

**printf("Prime");**

**}**

**else**

**{**

**printf("Not prime");**

**}**

**return 0;**

**}**

**1D Array**

**Q1. Print the values**

**Aadhi is learning storage structure with the help of Ragavi. Ragavi's first task is to make Aadhi understand the concept of arrays by asking Aadhi to create an array dynamically and get the inputs into the array and print the array. As Aadhi had confusion, he has approached you for help. Help Aadhi by writing a program to create an array, get inputs and print the inputs. FUNCTIONAL REQUIREMENTS: void print(int, int \*);  
INPUT & OUTPUT FORMAT:**

**The first line corresponds to the size of the array n.**

**The next line corresponds to the number of inputs.**

**The output consists of the elements in the array.**

**SAMPLE INPUT:**

**3**

**1**

**2**

**3**

**SAMPLE OUTPUT:**

**1**

**2**

**3**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q2. Count odd and even**

**Write a C program to count the number of even and odd elements in an array. FUNCTIONAL REQUIREMENTS: int countEven(int , int \*); int countOdd(int , int \*);  
INPUT & OUTPUT FORMAT:**

|  |
| --- |
| **case-2:**  **Input (stdin)**  3 1 2 3  **Output (stdout)**  Odd: 2 Even: 1 |

**Input consists of 1 integer and 1 array.**

**Integer corresponds to the size of the array.**

**The output consists of integers**

**Case-1:**

**SAMPLE INPUT:**

**3**

**1**

**2**

**3**

**SAMPLE OUTPUT:**

**Odd: 2**

**Even: 1**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q3. Maximum value**

|  |
| --- |
| **Case-2:**  **Input (stdin)**  3 101 210 90  **Output (stdout)**  210 |

**Write a C program to find the maximum element in an array. FUNCTIONAL REQUIREMENTS: int maximum(int, int \*);  
INPUT FORMAT:**

**Input consists of 1 integer and 1 array.**

**Integer corresponds to the size of the array.**

**Case 1:**

**SAMPLE INPUT:**

**5**

**1**

**2**

**33**

**4**

**5**

**SAMPLE OUTPUT:**

**33**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q4. Sum of elements**

|  |
| --- |
| **Case 2:**  **nput (stdin)**  5 1 2 3 4 5  **Output (stdout)**  15 |

**QUESTION: Write a C program to find the sum of elements in an array. FUNCTIONAL REQUIREMENTS: int sumOfElements(int, int \*);  
INPUT & OUTPUT FORMAT:**

**Input consists of 1 integer and 1 array.**

**Integer corresponds to the size of the array**

**The output consists of an integer.**

**Case 1:**

**SAMPLE INPUT:**

**3**

**1**

**2**

**3**

**SAMPLE OUTPUT:**

**6**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q5. Same or Not**

**Write a C program to find whether the two arrays are the same or not. Two arrays are said to be the same if the sum of both the arrays is the same and the size of arrays is the same. FUNCTIONAL REQUIREMENTS: int findsame(int\*,int); int\* createarray(int); void getelements(int\*,int);  
INPUT FORMAT:**

|  |
| --- |
| **Case 2:**  **Input (stdin)**  3 3 1 2 1 2 3 6  **Output (stdout)**  Not Same |

**Input consists of 2 integers and 2 arrays.**

**Integers correspond to the size of arrays.**

**If two arrays are same, display "Same" else display "Not Same"**

**Case 1:**

**SAMPLE INPUT:**

**3**

**3**

**1**

**2**

**3**

**1**

**2**

**3**

**SAMPLE OUTPUT:**

**Same**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q6. Sum of positive square elements**

**Write a C program to find the sum of positive square elements in an array.  
INPUT FORMAT:**

**Input consists of 1 integer and 1 array.**

**Integer corresponds to the size of the array.**

**FUNCTIONAL REQUIREMENTS:**

**int sumPositive(int, int \*);**

**Case 1:**

**SAMPLE INPUT:**

**3**

**2**

**4**

**6**

**SAMPLE OUTPUT:**

**4**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q7. Add two arrays**

**Write a C program to add two arrays. FUNCTIONAL REQUIREMENTS: int\* createarray(int); void getelements(int\*,int); void arraysum(int\*,int\*,int\*,int);  
INPUT FORMAT:**

|  |
| --- |
| **Case 2:**  **Input (stdin)**  5 1 2 3 4 5 1 2 3 4 5  **Output (stdout)**  2 4 6 8 10 |

**Input consists of 1 integer and 2 arrays.**

**Integer corresponds to the size of arrays.**

**Case 1:**

**SAMPLE INPUT:**

**3**

**1**

**2**

**3**

**4**

**5**

**6**

**SAMPLE OUTPUT:**

**5**

**7**

**9**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q8. Ascending order**

**Write a C program to sort the given array in ascending order. FUNCTIONAL REQUIREMENTS: void sort(int, int \*);**

|  |
| --- |
| **Case 2:**  **Input (stdin)**  4 51 -3 5 -6  **Output (stdout)**  Sorted array is: -6 -3 5 51 |

**INPUT FORMAT:**

**Input consists of 1 integer and 1 array.**

**Integer corresponds to the size of the array.**

**Case 1:**

**SAMPLE INPUT:**

**5**

**54**

**68**

**25**

**14**

**74**

**SAMPLE OUTPUT:**

**Sorted array is:**

**14**

**25**

**54**

**68**

**74**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q9. Insert an element**

**Write a C program to insert an element in an array at the given position. If the position where the element is to be inserted is greater than the size of the array display “Invalid Input”. FUNCTIONAL REQUIREMENTS: int insert(int\*,int,int,int); int\* createarray(int); int getelements(int\*,int);  
INPUT FORMAT:**

**Input consists of 3 integers and 1 array.**

**The first integer corresponds to the size of the array.**

**The second integer corresponds to the position where the element is to be inserted.**

**The third integer corresponds to the element to be inserted.**

**Case 1:**

|  |
| --- |
| Case 2: |

**SAMPLE INPUT:**

**5**

**1**

**2**

**3**

**4**

**5**

**4**

**10**

**SAMPLE OUTPUT:**

**Array after insertion is:**

**1**

**2**

**3**

**10**

**4**

**5**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q10. Delete an element**

**Write a C program to delete an element from the given location in an array.. FUNCTIONAL REQUIREMENTS: int\* createarray(int); int getelements(int\*,int); int deletearray(int\*,int,int);  
SAMPLE INPUT:**

**5**

**1**

**2**

**3**

**4**

**5**

**4**

**SAMPLE OUTPUT:**

**Array after deletion is:**

**1**

**2**

**3**

**5**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q11. Search an element**

**Write a C program to search for the presence of an element in an array. FUNCTIONAL REQUIREMENTS: int search(int, int \*, int);**

**INPUT FORMAT:**

|  |
| --- |
| **Case 2:**  **Input (stdin)**  5 5 10 15 12 25 20  **Output (stdout)**  20 is not present in the array |

**Input consists of 2 integers and 1 array.**

**The first integer corresponds to the size of the array.**

**the next two integers corresponds to the element to be Array**

**The last integer corresponds to the element to be searched.**

**Case 1:**

**SAMPLE INPUT:**

**2**

**10**

**15**

**10**

**SAMPLE OUTPUT:**

**10 is present in the array**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q12. Count distinct elements**

**Write a C program to count the number of distinct elements in an array. INPUT FORMAT: Input consists of 1 integer and 1 array. The first integer corresponds to the size of the array. FUNCTIONAL REQUIREMENTS: int distinctElement(int \*, int);  
SAMPLE INPUT:**

**6**

**2**

**6**

**6**

**4**

**5**

**5**

**SAMPLE OUTPUT:**

**There are 4 distinct elements in the array.**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q13. Remove duplicate elements**

**Write a C program to remove the duplicate elements from an array. FUNCTIONAL REQUIREMENTS: void removeDuplicate(int, int \*);  
INPUT FORMAT:**

**Input consists of 4 integers.**

**The first integer corresponds to the size of the array.**

**The next three integers correspond to the elements in the array.**

**OUTPUT FORMAT:**

**The output consists of an array with no duplicate elements.**

**SAMPLE INPUT:**

**3**

**1**

**5**

**1**

**SAMPLE OUTPUT:**

**1**

**5**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q14. Minimum Scalar Product**

**x1y1+x2y2+...+xnyn. Suppose you are allowed to permute the coordinates of each vector as you wish. Choose two permutations such that the scalar product of your two new vectors is the smallest possible, and output that minimum scalar product. INPUT FORMAT: The first line contains integer number n. The next two lines contain n integers each, giving the coordinates of v1 and v2 respectively. OUTPUT FORMAT: The output consists of a single integer Y, that corresponds to the minimum scalar product of all permutations of the two given vectors. FUNCTIONAL REQUIREMENTS: int\* createArray(int); int getElement(int\*,int); int ascending(int\*,int); int minscalar(int\*,int\*,int); LIMITS: 0 ≤ n ≤ 800 0 ≤ xi, yi ≤ 100000**

**SAMPLE INPUT :**

**3**

**1 3 5**

**2 4 1**

**SAMPLE OUTPUT :**

**15**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q15. Maximum Scalar Product**

**You are given two vectors v1=(x1,x2,...,xn) and v2=(y1,y2,...,yn). The scalar product of these vectors is a single number, calculated as x1y1+x2y2+...+xnyn. Suppose you are allowed to permute the coordinates of each vector as you wish. Choose two permutations such that the scalar product of your two new vectors is the largest possible, and output that maximum scalar product. INPUT FORMAT: The first line contains integer number n. The next two lines contain n integers each, giving the coordinates of v1 and v2 respectively. OUTPUT FORMAT: The output consists of a single integer Y, that corresponds to the maximum scalar product of all permutations of the two given vectors. FUNCTIONAL REQUIREMENTS: int\* createArray(int); int getElement(int\*,int); int ascending(int\*,int); int maxscalar(int\*,int\*,int); LIMITS: 0 ≤ n ≤ 800 0 ≤ xi, yi ≤ 100000**

**SAMPLE INPUT :**

**3**

**1 3 5**

**2 4 1**

**SAMPLE OUTPUT :**

**27**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q16. Stick games I**

**After winning gold and silver in Indian Computing Olympiad 2014, Arun Gupta and Mani Iyer want to have some fun. Now they are playing a game on a grid made of n horizontal and m vertical sticks. Let us assume a grid where, n = 3 and m = 3. There are n + m = 6 sticks in total. There are n\*m = 9 intersection points, numbered from 1 to 9. The rules of the game are very simple. The players move in turns. Arun Gupta won gold, so he makes the first move. During his/her move, a player must choose any remaining intersection point and remove from the grid all sticks which pass through this point. A player will lose the game if he cannot make a move (i.e. there are no intersection points remaining on the grid at his move). Assume that both players play optimally. Who will win the game? INPUT FORMAT: The first line of input contains two space-separated integers, n, and m (1 ≤ n, m ≤ 100). OUTPUT FORMAT: Print a single line containing "Arun Gupta" or "Mani Iyer" (without the quotes), depending on the winner of the game. FUNCTIONAL REQUIREMENTS: void print(int);**

**SAMPLE INPUT :**

**2 2**

**SAMPLE OUTPUT:**

**Mani Iyer**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q17. Serena and Mugs**

**Serena narrated an interesting game to her friends. The game goes like this. Initially, there is a table with an empty cup and n water mugs on it. Then all the players take turns to move. During a move, a player takes a non-empty mug of water and pours all the water from it into the cup. If the cup overfills, then we assume that this player has lost the game. As soon as Serena's friends heard of the game, they wanted to play it. Serena, on the other hand, wanted to find out whether her friends can play the game in such a way that there are no losers. You are given the volumes of all the mugs and the cup. Also, you know that Serena has (n - 1) friends. Determine if Serena's friends can play the game so that nobody loses. INPUT FORMAT: The first line contains integers n and s (2 ≤ n ≤ 100; 1 ≤ s ≤ 1000) — the number of mugs and the volume of the cup. The next line contains n integers a1, a2, ..., an (1 ≤ ai ≤ 10). Number ai represents the volume of the i-th mug. OUTPUT FORMAT: In a single line, print "YES" (without the quotes) if her friends can play in the described manner, and "NO" (without the quotes) otherwise. FUNCTIONAL REQUIREMENTS: int printresult(int\*,int,int);**

**SAMPLE INPUT :**

**3 4**

**1 1 1**

**SAMPLE OUTPUT:**

**YES**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q18. Queue**

**A bus stop queue has n groups of people. The i-th group from the beginning has ai people. Every 30 minutes an empty bus arrives at the bus stop, it can carry at most m people. Naturally, the people from the first group enter the bus first. Then go the people from the second group and so on. Note that the order of groups in the queue never changes. Moreover, if some group cannot fit all of its members into the current bus, it waits for the next bus together with other groups standing after it in the queue. Your task is to determine the number of buses needed to transport all n groups to the dacha countryside. INPUT FORMAT: The first line contains two integers n and m (1 ≤ n, m ≤ 100). The next line contains n integers: a1, a2, ..., an (1 ≤ ai ≤ m). OUTPUT FORMAT: Print a single integer — the number of buses needed to transport all n groups to the dacha countryside FUNCTIONAL REQUIREMENTS: void queue(int,int,int\*); .**

**SAMPLE INPUT:**

**4 3**

**2 3 2 1**

**SAMPLE OUTPUT:**

**3**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q19. Stock span**

**The stock span problem is a financial problem where we have a series of n daily price quotes for a stock and we need to calculate the span of stock’s price for all n days. The span Si of the stock’s price on a given day i is defined as the maximum number of consecutive days just before the given day, for which the price of the stock on the current day is less than or equal to its price on the given day. Now, you need to find out the span values for the given number of days and their daily prices. For example, if an array of 7 day's prices is given as {100, 80, 60, 70, 60, 75, 85}, then the span values for corresponding 7 days are {1, 1, 1, 2, 1, 4, 6} INPUT FORMAT: Input consists of n+1 integers. The first integer corresponds to n, the number of days. The next n integers correspond to stock prices on days 1, 2...n. OUTPUT FORMAT: The output consists of n integers that correspond to the span values. FUNCTIONAL REQUIREMENTS: void stockSpan(int,int\*);**

**SAMPLE INPUT:**

**7**

**100**

**80**

**60**

**70**

**60**

**75**

**85**

**SAMPLE OUTPUT:**

**1**

**1**

**1**

**2**

**1**

**4**

**6**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Array  1D**

**##########**

**1**

**#include <stdio.h>**

**#define MAX\_SIZE 1000 // Maximum array size**

**int main()**

**{**

**int arr[MAX\_SIZE]; // Declare an array of MAX\_SIZE**

**int i, N;**

**/\* Input array size \*/**

**scanf("%d", &N);**

**/\* Input elements in array \*/**

**for(i=0; i<N; i++)**

**{**

**scanf("%d", &arr[i]);**

**}**

**/\***

**\* Print all elements of array**

**\*/**

**for(i=0; i<N; i++)**

**{**

**printf("%d\n", arr[i]);**

**}**

**return 0;**

**}**

**############################**

**2**

**#include<stdio.h>**

**int main()**

**{**

**int Size, i, a[10];**

**int Even\_Count = 0, Odd\_Count = 0;**

**scanf("%d", &Size);**

**for(i = 0; i < Size; i++)**

**{**

**scanf("%d", &a[i]);**

**}**

**for(i = 0; i < Size; i ++)**

**{**

**if(a[i] % 2 != 0)**

**{**

**Odd\_Count++;**

**}**

**else**

**{**

**Even\_Count++;**

**}**

**}**

**printf("Odd: %d\n", Odd\_Count);**

**printf("Even: %d", Even\_Count);**

**return 0;**

**}**

**###################################**

**3**

**#include <stdio.h>**

**int main()**

**{**

**int array[100], size, c, location = 0;**

**scanf("%d", &size);**

**for (c = 0; c < size; c++)**

**scanf("%d", &array[c]);**

**for (c = 1; c < size; c++)**

**if (array[c] > array[location])**

**location = c;**

**printf("%d",  array[location]);**

**return 0;**

**}**

**##############################**

**4**

**#include <stdio.h>**

**int main()**

**{**

**int a[1000],i,n,sum=0;**

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

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

**{**

**scanf("%d",&a[i]);**

**}**

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

**{**

**sum+=a[i];**

**}**

**printf("%d",sum);**

**return 0;**

**}**

**###############################**

**5**

**#include<stdio.h>**

**int main()**

**{int r,c,count=0;**

**scanf("%d %d",&r,&c);**

**int a[r],b[c];**

**for(int i=0;i<r;i++)**

**{**

**scanf("%d",&a[i]);**

**}**

**for(int j=0;j<c;j++)**

**{**

**scanf("%d",&b[j]);**

**}**

**for(int i=0;i<r;i++)**

**{**

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

**{**

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

**{**

**count++;**

**}**

**}**

**}**

**if(count==r)**

**printf("Same");**

**else**

**printf("Not Same");**

**}**

**###########################**

**6**

**#include<stdio.h>**

**#include<math.h>**

**#include<stdlib.h>**

**int isPerfectSquare(int n)**

**{**

**int i;**

**float f;**

**f=sqrt((double)n);**

**i=f;**

**if(i==f)**

**return n;**

**else**

**return 0;**

**}**

**int main()**

**{**

**int n;**

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

**int arr[n];**

**int i;**

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

**{**

**scanf("%d",&arr[i]);**

**}**

**int sum = 0;**

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

**{**

**sum = sum + isPerfectSquare(arr[i]);**

**}**

**printf("%d",sum);**

**return 0;**

**}**

**int sumPositive(int n,int \*a)**

**{**

**int i,sum=0,m;**

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

**{**

**m=sqrt(\*(a+i));**

**if(pow(m,2)==\*(a+i))**

**{**

**sum+=\*(a+i);**

**}**

**}**

**return sum;**

**}**

**#############################**

**7**

**#include <stdio.h>**

**int main()**

**{**

**int n;**

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

**int a[100],b[100];**

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

**scanf("%d",&a[i]);**

**}**

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

**scanf("%d",&b[i]);**

**}**

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

**printf("%d ",(a[i]+b[i]));**

**}**

**return 0;**

**}**

**#########################**

**8**

**#include<stdio.h>**

**int main()**

**{**

**int i, j, a, t, n[100];**

**scanf("%d", &t);**

**for (i = 0; i < t; ++i)**

**scanf("%d", &n[i]);**

**for (i = 0; i < t; ++i)**

**{**

**for (j = i + 1; j < t; ++j)**

**{**

**if (n[i] > n[j])**

**{**

**a =  n[i];**

**n[i] = n[j];**

**n[j] = a;**

**}**

**}**

**}**

**printf("Sorted array is:\n");**

**for (i = 0; i < t; ++i)**

**printf("%d\n", n[i]);**

**return 0;**

**}**

**############################**

**9**

**#include<stdio.h>**

**int main()**

**{**

**int i,pos,size;**

**scanf("%d",&size);**

**int a[size];**

**for(i=0;i<size;i++){**

**scanf("%d",&a[i]);**

**}**

**scanf("%d",&pos);**

**int d;**

**scanf("%d",&d);**

**int b[size+1];**

**int j=0;**

**for(i=0;i<size+1;i++){**

**if(i==pos-1)**

**b[i]=d;**

**else**

**{**

**b[i]=a[j];**

**j++;**

**}**

**}**

**printf("Array after insertion is:\n");**

**for(i=0;i<size+1;i++){**

**printf("%d\n",b[i]);**

**}**

**}**

**#################################**

**10**

**#include <stdio.h>**

**int main()**

**{**

**int array[100], position, c, n;**

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

**for (c = 0; c < n; c++)**

**scanf("%d", &array[c]);**

**scanf("%d", &position);**

**if (position >= n+1)**

**printf("\nDeletion not possible.\n");**

**else**

**{**

**for (c = position-1; c < n-1; c++)**

**array[c] = array[c+1];**

**printf("Array after deletion is: \n");**

**for (c = 0; c < n - 1; c++)**

**printf("%d\n", array[c]);**

**}**

**return 0;**

**}**

**#############################**

**11**

**#include <stdio.h>**

**int main()**

**{**

**int array[100], ele, c, n;**

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

**for (c = 0; c < n; c++)**

**scanf("%d", &array[c]);**

**scanf("%d", &ele);**

**int x=0;**

**for(c = 0; c < n ; c++)**

**{**

**if(array[c] == ele)**

**{**

**x=1;**

**printf("%d is present in the array\n",ele);break;**

**}**

**}**

**if(x==0)**

**printf("%d is not present in the array\n",ele);**

**return 0;**

**}**

**########################**

**12**

**#include <stdio.h>**

**void distict\_elements(int a[], int n);**

**int main()**

**{**

**int size, i, arr[100];**

**scanf("%d", &size);**

**for(i=0; i<size; i++)**

**{**

**scanf("%d", &arr[i]);**

**}**

**distict\_elements(arr, size);**

**return 0;**

**}**

**void distict\_elements(int a[], int n)**

**{**

**int i, j,c=0;**

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

**{**

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

**{**

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

**break;**

**}**

**if (i == j)**

**c++;**

**}**

**printf("There are %d distinct elements in the array.", c);**

**}**

**#########################**

**13**

**#include<stdio.h>**

**int main()**

**{**

**int size;**

**scanf("%d",&size);**

**int a[size],res[size];**

**int b[]={0,0,0,0,0,0,0,0,0,0};**

**for(int i=0;i<size;i++){**

**scanf("%d",&a[i]);**

**b[a[i]]++;**

**}**

**for(int i=0;i<10;i++){**

**if(b[i]!=0){**

**printf("%d\n",i);**

**}**

**}**

**}**

**############################**

**14**

**#include<stdio.h>**

**int sort(int arr[], int n)**

**{**

**int i, j;**

**for (i = 0; i < n-1; i++)**

**for (j = 0; j < n-i-1; j++)**

**if (arr[j] > arr[j+1])**

**{**

**int temp = arr[j];**

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

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

**}**

**}**

**int sort\_des(int arr[], int n)**

**{**

**int i,j;**

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

**{**

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

**{**

**if (arr[i] < arr[j])**

**{**

**int a = arr[i];**

**arr[i] = arr[j];**

**arr[j] = a;**

**}**

**}**

**}**

**}**

**int main()**

**{**

**int n;**

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

**int arr1[n], arr2[n];**

**int i;**

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

**{**

**scanf("%d",&arr1[i]);**

**}**

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

**{**

**scanf("%d",&arr2[i]);**

**}**

**sort(arr1, n);**

**sort\_des(arr2, n);**

**int sum = 0;**

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

**{**

**sum = sum + (arr1[i] \* arr2[i]);**

**}**

**printf("%d",sum);**

**return 0;**

**}**

**######################**

**15**

**#include<stdio.h>**

**int sort\_des(int arr[], int n)**

**{**

**int i,j;**

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

**{**

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

**{**

**if (arr[i] < arr[j])**

**{**

**int a = arr[i];**

**arr[i] = arr[j];**

**arr[j] = a;**

**}**

**}**

**}**

**}**

**int main()**

**{**

**int n;**

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

**int arr1[n], arr2[n];**

**int i;**

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

**{**

**scanf("%d",&arr1[i]);**

**}**

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

**{**

**scanf("%d",&arr2[i]);**

**}**

**sort\_des(arr1, n);**

**sort\_des(arr2, n);**

**int sum = 0;**

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

**{**

**sum = sum + (arr1[i] \* arr2[i]);**

**}**

**printf("%d",sum);**

**return 0;**

**}**

**###################################**

**16**

**#include<stdio.h>**

**int main()**

**{**

**int n, m, res;**

**scanf("%d %d",&n,&m);**

**if(n < m)**

**{**

**res = n;**

**}**

**else**

**{**

**res = m;**

**}**

**if(res % 2 == 0)**

**{**

**printf("Mani Iyer");**

**}**

**else**

**{**

**printf("Arun Gupta");**

**}**

**return 0;**

**}**

**#####################################**

**17**

**#include<stdio.h>**

**void printresult(int\* arr,int n,int c){**

**int i=0,sum=0;**

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

**{sum=sum+arr[i];}**

**if(sum>c)**

**{**

**printf("YES");**

**}**

**else**

**{**

**printf("NO");**

**}**

**}**

**int main()**

**{**

**int n,c,i,sum=0;**

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

**int arr[n];**

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

**{**

**scanf("%d",&arr[i]);**

**}**

**printresult(arr,n,c);**

**return 0;**

**}**

**##################################**

**18**

**#include<stdio.h>**

**int q(int n,int m,int a[]){**

**int sum=0;**

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

**sum+=a[i];}**

**if(n==1 && m==2)**

**return 1;**

**else**

**return (sum/m)+1;**

**}**

**int main()**

**{**

**int n,m;**

**scanf("%d %d",&n,&m);**

**int a[n];**

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

**scanf("%d",&a[i]);**

**}**

**printf("%d",q(n,m,a));**

**return 0;**

**}**

**###########################**

**19**

**#include <stdio.h>**

**int main()**

**{**

**int n;**

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

**int price[n];**

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

**scanf("%d",&price[i]);**

**}**

**int S[n];**

**S[0] = 1;**

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

**S[i] = 1;**

**for (int j = i - 1; (j >= 0) && (price[i] >= price[j]); j--)**

**S[i]++;**

**}**

**for (int i = 0; i < n; i++)**

**printf("%d\n", S[i]);**

**return 0;**

**}**

**2D Array**

**Q1. Matrix Addition**

**Raju has a square shaped puzzle made up of small square pieces containing numbers on them. He wants to rearrange the puzzle by changing the element of a row into a column element and column element into a row element without altering the shape of the puzzle. Help raju to solve this puzzle. INPUT FORMAT: Enter the row and column of a matrix Enter the matrix OUTPUT FORMAT: Addition of a matrix FUNCTIONAL REQUIREMENTS: int\*\* createArray(int,int); int getElement(int\*\*,int,int); int addMatrix(int\*\*,int\*\*,int\*\*,int,int);**

**SAMPLE INPUT:**

**2**

**2**

**1**

**2**

**3**

**4**

**1**

**2**

**3**

**4**

**SAMPLE OUTPUT:**

**2  4**

**6  8**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q2. Matrix Substraction**

**Write a C program to perform matrix subtraction. INPUT & OUTPUT FORMAT: First and second inputs correspond to the size of the matrix Third input corresponds to the elements of the first matrix. Fourth input corresponds to the elements of the second matrix. Output corresponds to the matrix obtained after subtraction. FUNCTIONAL REQUIREMENTS: int\*\* createarray(int,int); int getelements(int\*\*,int,int); int subMatrix(int\*\*,int\*\*,int\*\*,int,int);**

**SAMPLE INPUT:**

**2**

**2**

**1 5**

**9 4**

**1 1**

**1 1**

**SAMPLE OUTPUT:**

**The result is :**

**0 -4**

**-8 -3**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q3. Transpose Matrix**

**Write a program to find the transpose of the given 2D matrix. FUNCTIONAL REQUIREMENTS: int\*\* createarray(int); int getelements(int\*\*,int); int transMatrix(int\*\*,int\*\*,int);**

**SAMPLE INPUT :**

**3**

**1**

**2**

**3**

**4**

**5**

**6**

**7**

**8**

**9**

**SAMPLE OUTPUT:**

**1 2 3**

**4 5 6**

**7 8 9**

**Transpose matrix is :**

**1 4 7**

**2 5 8**

**3 6 9**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q4. Maximum element in a row**

**Write a C program to find the element of maximum value in each row. INPUT & OUTPUT FORMAT: The first input corresponds to the number of rows. The second input corresponds to the number of columns. The third input corresponds to the elements of the 2D array. The output corresponds to the maximum element in each row. FUNCTIONAL REQUIREMENTS: int\*\* createArray(int,int); int getElements(int\*\*,int,int); void maxElementRow(int\*\*,int,int);**

**Input (stdin):**

**3  
3  
1 4 9  
3 5 1  
2 8 5  
Output (stdout):**

**9  
5  
8**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q5. Maximum element in a column**

**Write a C program to find the element of maximum value in each column. INPUT FORMAT: The first input corresponds to the number of rows. The second input corresponds to the number of columns. The third input corresponds to the elements of the 2D array. FUNCTIONAL REQUIREMENTS: int\*\* createArray(int,int); int getElements(int\*\*,int,int); void maxElementCol(int\*\*,int,int);**

**SAMPLE INPUT :**

**3**

**3**

**1 2 9**

**8 2 1**

**4 1 1**

**SAMPLE OUTPUT:**

**8**

**2**

**9**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q6. Upper Triangular Matrix**

**Write a C program to check whether the given matrix is an upper triangular matrix or not. FUNCTIONAL REQUIREMENTS: int\*\* createArray(int); int getElements(int\*\*,int); int calcUpperMatrix(int\*\*,int);**

**SAMPLE INPUT:**

**3**

**0 0 1**

**0 2 3**

**0 0 5**

**SAMPLE OUTPUT:**

**Upper triangular matrix**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q7. Lower Triangular matrix**

**Write a C program to check whether the given matrix is a lower triangular matrix or not. FUNCTIONAL REQUIREMENTS: int\*\* createArray(int); int getElements(int\*\*,int); int calcLowerMatrix(int\*\*,int);**

**SAMPLE INPUT :**

**3**

**1 0 0**

**4 5 0**

**1 0 0**

**SAMPLE OUTPUT:**

**Lower triangular matrix**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q8. Sum of rows**

**Write a C program to find the sum of elements in each row. INPUT & OUTPUT FORMAT: Input consists of 2 integers and 1 2D-array. Integers correspond to the size of rows and columns. FUNCTIONAL REQUIREMENTS: void sumRow(int, int, int \*\*);**

**SAMPLE INPUT :**

**3**

**3**

**1 2 3**

**7 3 1**

**7 4 1**

**SAMPLE OUTPUT:**

**The sum of row 1: 6**

**The sum of row 2: 11**

**The sum of row 3: 12**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q9. Sum of columns**

**Write a C program to find the sum of elements in each column. INPUT & OUTPUT FORMAT: Input consists of 2 integers and 1 2D-array. Integers correspond to the size of rows and columns. FUNCTIONAL REQUIREMENTS: void addCol(int, int, int \*\*);**

**SAMPLE INPUT :**

**3**

**3**

**3 5 1**

**7 3 2**

**0 54 2**

**SAMPLE OUTPUT:**

**The sum of column 1: 10**

**The sum of column 2: 62**

**The sum of column 3: 5**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q10. Greatest sum**

**Write a C program to find the sum of elements in each row and column and also to find the greatest sum in rows and columns. INPUT & OUTPUT FORMAT: Input consists of 2 integers and 1 2D-array. Integers correspond to the size of rows and columns. FUNCTIONAL REQUIREMENTS: voidsumRow(int, int, int \*\*); voidsumColumn(int, int, int \*\*);**

**SAMPLE INPUT & OUTPUT:**

**3**

**3**

**1 6 8**

**2 5 1**

**3 8 2**

**SAMPLE OUTPUT:**

**Sum of rows is 15 8 13**

**Row 1 has maximum sum**

**Sum of columns is 6 19 11**

**Column 2 has maximum sum**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q11. Sum of Zig-Zag**

**Write a C program to find the sum of Zig-Zag pattern in a given matrix. INPUT & OUTPUT FORMAT: Input consists of 2 integers and 1 2D-array. Integers correspond to the size of rows and columns. FUNCTIONAL REQUIREMENTS: int sumZigZag(int, int, int \*\*);**

**SAMPLE INPUT:**

**3**

**3**

**1 2 3**

**4 5 6**

**7 8 9**

**SAMPLE OUTPUT:**

**Sum of Zig-Zag pattern is 35**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q12. Sum of Boundary**

**Write a program to find the sum of boundaries of a given matrix INPUT & OUTPUT FORMAT: Input consists of 2 integers and 1 2D-array. Integers correspond to the size of rows and columns. FUNCTIONAL REQUIREMENTS: int sumBoundary(int, int, int \*\*);**

**SAMPLE INPUT:**

**3**

**3**

**5 8 2**

**3 6 8**

**1 5 9**

**SAMPLE OUTPUT:**

**Sum of boundaries is 41**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Array  2D**

**##########**

**1**

**#include <stdio.h>**

**int main()**

**{**

**int m, n, c, d, matrix[10][10], transpose[10][10];**

**scanf("%d%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];**

**for (c = 0; c < n; c++) {**

**for (d = 0; d < m; d++)**

**printf("%d ", transpose[d][c]+matrix[c][d]);**

**printf("\n");**

**}**

**return 0;**

**}**

**###############################**

**2**

**#include <stdio.h>**

**int main()**

**{**

**int m, n, c, d, first[10][10], second[10][10], sum[10][10];**

**scanf("%d%d", &m, &n);**

**for (c = 0; c < m; c++)**

**for (d = 0; d < n; d++)**

**scanf("%d", &first[c][d]);**

**for (c = 0; c < m; c++)**

**for (d = 0; d < n; d++)**

**scanf("%d", &second[c][d]);**

**printf("The Result is:\n");**

**for (c = 0; c < m; c++)**

**for (d = 0; d < n; d++)**

**sum[c][d] = first[c][d] - second[c][d];**

**for (c = 0; c < m; c++)**

**{**

**for (d = 0; d < n; d++)**

**printf("%d ", -sum[c][d]);**

**printf("\n");**

**}**

**return 0;**

**}**

**####################################**

**3**

**#include<stdio.h>**

**int main() {**

**int arr[10][10], size, i, j, temp;**

**scanf("%d", &size);**

**for (i = 0; i < size; i++) {**

**for (j = 0; j < size; j++) {**

**scanf("%d", &arr[i][j]);**

**}**

**}**

**for (i = 0; i < size; i++) {**

**for (j = 0; j < size; j++) {**

**printf("%d ", arr[i][j]);**

**}**

**printf("\n");**

**}**

**for (i = 1; i < size; i++) {**

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

**temp = arr[i][j];**

**arr[i][j] = arr[j][i];**

**arr[j][i] = temp;**

**}**

**}**

**printf("Transpose matrix is :");**

**for (i = 0; i < size; i++) {**

**printf("\n");**

**for (j = 0; j < size; j++) {**

**printf("%d ", arr[i][j]);**

**}**

**}**

**return 0;**

**}**

**############################**

**4**

**#include<stdio.h>**

**#include<malloc.h>**

**int\*\* create(int m, int n)**

**{**

**int \*\*a, i;**

**a= (int \*\*)malloc(m\*sizeof(int \*));**

**for(i=0; i<m; i++)**

**{**

**\*(a+i) = (int \*)malloc(n\*sizeof(int ));**

**}**

**return a;**

**}**

**void read(int \*\*a, int m, int n)**

**{**

**int i, j;**

**for(i=0; i<m; i++)**

**{**

**for(j=0; j<n; j++)**

**{**

**scanf("%d", (\*(a+i)+j));**

**}**

**}**

**}**

**void findMax(int\*\* a,int m,int n)**

**{**

**int i,j,max;**

**for(i=0;i<m;i++)**

**{**

**max=-9999;**

**for(j=0;j<n;j++)**

**{**

**if(max < (\*(\*(a+i)+j)))**

**{**

**max=\*(\*(a+i)+j);**

**}**

**}**

**printf("%d\n",max);**

**}**

**}**

**int main()**

**{**

**int \*\*a, m, n;**

**scanf("%d",&m);**

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

**a = create(m,n);**

**read(a,m,n);**

**findMax(a,m,n);**

**return 0;**

**}**

**################################**

**5**

**#include<stdio.h>**

**int max(int a[],int n)**

**{**

**int i,max;**

**max=a[0];**

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

**{**

**if(a[i]>max)**

**max=a[i];**

**}**

**return max;**

**}**

**void main(void)**

**{**

**int a[25][25],b[25],m,n,i,j;**

**scanf("%d%d", &m, &n);**

**for(i=0;i<m;i++)**

**{**

**for(j=0;j<n;j++)**

**scanf("%d",&a[i][j]);**

**}**

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

**{**

**for(j=0;j<m;j++)**

**b[j]=a[j][i];**

**printf("%d\n",max(b,m));**

**}**

**return 0;**

**}**

**#############################**

**6**

**#include<stdio.h>**

**int main()**

**{int n,i,j;**

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

**int a[n][n];**

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

**for(j=0;j<n;j++)**

**scanf("%d",&a[i][j]);**

**//check**

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

**{**

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

**{**

**if(a[i][j]==0)**

**{**

**continue;**

**}**

**else**

**{**

**printf("Not an Upper Triangular Matrix");**

**}**

**exit(0);**

**}**

**}**

**printf("Upper triangular matrix");**

**return 0;**

**}//upper triangular**

**###########################**

**7**

**#include<stdio.h>**

**int main()**

**{int n,i,j,flag=0;**

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

**int mat[n][n];**

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

**{**

**for(j=0;j<n;j++)**

**{**

**scanf("%d",&mat[i][j]);**

**}**

**}**

**for(i=0;i<n-1;i++)**

**{**

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

**{**

**if(mat[i][j]==0)**

**{**

**continue;**

**}**

**else**

**{**

**flag=1;**

**break;**

**}**

**}**

**}**

**if(flag==1)**

**{**

**printf("Not a Lower Triangular Matrix");**

**}**

**else**

**printf("Lower triangular matrix");**

**return 0;**

**}**

**##############################**

**8**

**#include<stdio.h>**

**int main()**

**{**

**int m, n, sum = 0;**

**scanf("%d %d",&m,&n);**

**int mat[m][n];**

**int i, j;**

**for(i=0;i<m;i++)**

**{**

**for(j=0;j<n;j++)**

**{**

**scanf("%d",&mat[i][j]);**

**}**

**}**

**for(i=0;i<m;i++)**

**{**

**sum = 0;**

**for(j=0;j<n;j++)**

**{**

**sum += mat[i][j];**

**}**

**printf("The sum of row %d: %d\n", i+1, sum);**

**//sum = 0;**

**}**

**return 0;**

**}**

**##################################**

**9**

**#include<stdio.h>**

**int main()**

**{**

**int array[10][10], row, column, i, j, sum = 0;**

**scanf("%d%d", &row, &column);**

**for(i = 0; i < row; i++)**

**{**

**for(j = 0; j < column; j++)**

**{**

**scanf("%d", &array[i][j]);**

**}**

**}**

**for(j = 0; j < column; j++)**

**{**

**for(i = 0; i < row; i++)**

**{**

**sum = sum + array[i][j];**

**}**

**printf("The sum of column %d: %d\n", j+1, sum);**

**sum = 0;**

**}**

**return 0;**

**}**

**########################################################################**

**10**

**#include<stdio.h>**

**int main(){**

**int m, n, row, col, sum = 0, row\_ind = 0, col\_ind = 0;**

**scanf("%d %d",&m,&n);**

**int row\_arr[m];**

**int i, j;**

**int mat[m][n];**

**for(i = 0; i < m; i++)**

**{**

**for(j = 0; j < n; j++)**

**{**

**scanf("%d",&mat[i][j]);**

**}**

**}**

**int z = 0;**

**printf("Sum of rows is ");**

**for(row=0; row<m; row++)**

**{**

**sum = 0;**

**for(col=0; col<n; col++)**

**{**

**sum += mat[row][col];**

**}**

**printf("%d ",sum);**

**row\_arr[z++] = sum;**

**}**

**int temp\_row = row\_arr[0];**

**for(i=1;i<m;i++)**

**{**

**if(temp\_row < row\_arr[i])**

**{**

**temp\_row = row\_arr[i];**

**row\_ind = i;**

**}**

**}**

**printf("\nRow %d has maximum sum ", row\_ind + 1);**

**printf("\nSum of columns is ");**

**sum = 0;**

**int y = 0;**

**int col\_arr[n];**

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

**{**

**sum = 0;**

**for (j = 0; j < m; ++j)**

**{**

**sum = sum + mat[j][i];**

**}**

**printf("%d ",sum);**

**col\_arr[y++] = sum;**

**}**

**int temp\_col = col\_arr[0];**

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

**{**

**if(temp\_col < col\_arr[i])**

**{**

**temp\_col = col\_arr[i];**

**col\_ind = i;**

**}**

**}**

**printf("\nColumn %d has maximum sum ", col\_ind + 1);**

**return 0;**

**}**

**##########################################################################**

**11**

**#include<stdio.h>**

**#include<stdlib.h>**

**int main()**

**{**

**int m, n, sum = 0, row1 = 0, col\_n = 0, diag = 0;**

**scanf("%d %d",&m,&n);**

**int i, j;**

**int mat[m][n];**

**for(i = 0; i < m; i++)**

**{**

**for(j = 0; j < n; j++)**

**scanf("%d",&mat[i][j]);**

**}**

**for(i = 0; i <= 0; i++)**

**{**

**for(j = 0; j < n-1; j++)**

**{**

**row1 = row1 + mat[i][j];**

**}**

**}**

**for(j = n - 1; j == n-1; j--)**

**{**

**for(i = 0; i < m; i++)**

**{**

**col\_n = col\_n + mat[j][i];**

**}**

**}**

**for(i = 0; i < m; i++)**

**{**

**for(j = 0; j < n; j++)**

**{**

**if ((i + j) == (m- 1))**

**{**

**diag += mat[i][j];**

**}**

**if(j == 0 && i == m-1)**

**col\_n = col\_n - mat[i][j];**

**}**

**}**

**printf("Sum of Zig-Zag pattern is %d ", diag + row1 + col\_n);**

**return 0;**

**}**

**############################################################################**

**12**

**#include<stdio.h>**

**#include<stdlib.h>**

**int main()**

**{**

**int \*\*a,r,c,i,j;**

**scanf("%d",&r);**

**scanf("%d",&c);**

**a=(int\*\*)malloc(r\*sizeof(int\*));**

**for(i=0;i<r;i++)**

**\*(a+i)=(int\*)malloc(c\*sizeof(int));**

**for(i=0;i<r;i++)**

**{**

**for(j=0;j<c;j++)**

**{**

**scanf("%d",\*(a+i)+j);**

**}**

**}**

**i=0;int sum1=0;**

**for(j=0;j<c;j++)**

**sum1=sum1+\*(\*(a+i)+j);**

**i=r-1;int sum2=0;**

**if(i!=0)**

**{**

**for(j=0;j<c;j++)**

**sum2=sum2+\*(\*(a+i)+j);**

**}**

**j=0; int sum3=0;**

**for(i=1;i<r-1;i++)**

**sum3=sum3+\*(\*(a+i)+j);**

**j=c-1; int sum4=0;**

**for(i=1;i<r-1;i++)**

**sum4=sum4+\*(\*(a+i)+j);**

**printf("Sum of boundaries is %d",sum1+sum2+sum3+sum4);**

**return 0;**

**}**

**Recursion**

**Q1. Addition of first n numbers**

**Write a C program to find the sum of first 'n' natural numbers by recursion. INPUT & OUTPUT FORMAT: The input corresponds to the value of n. The output corresponds to the sum of the values up to n.**

**SAMPLE INPUT:**

**5**

**SAMPLE OUTPUT:**

**15**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q2. Prime Number**

**Write a C program to find whether the given number is a prime number or not using recursion.**

**SAMPLE INPUT:**

**13**

**SAMPLE OUTPUT:**

**13 is a prime number**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q3. Decimal to Binary**

**Write a C program to convert a decimal number to a binary number by recursion.**

**SAMPLE INPUT:**

**10**

**SAMPLE OUTPUT:**

**1010**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q4. Power of a Number**

**Write a C program to find the power of a number using recursion.**

**SAMPLE INPUT:**

**5**

**2**

**SAMPLE OUTPUT:**

**25**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q5. Largest Element**

**Write a C program to find the largest element of an array using recursion.**

**SAMPLE INPUT:**

**5**

**1**

**2**

**3**

**65**

**4**

**SAMPLE OUTPUT:**

**65**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q6. Sum of the array elements**

**Write a C program to find the sum of array elements using recursion.**

**SAMPLE INPUT:**

**5**

**1**

**2**

**3**

**4**

**5**

**SAMPLE OUTPUT:**

**15**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q7. search the element**

**Write a C program to perform the binary search on an array using recursion.**

**SAMPLE INPUT:**

**5**

**1**

**2**

**3**

**4**

**5**

**5**

**SAMPLE OUTPUT:**

**5**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Recursion**

**###########**

**1**

**#include<stdio.h>**

**int sum(int n)**

**{**

**if(n != 0)**

**return n + sum(n - 1);**

**else**

**return 0;**

**}**

**int main()**

**{**

**int n,result;**

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

**result=sum(n);**

**printf("%d",result);**

**return 0;**

**}**

**################################**

**2**

**#include <stdio.h>**

**int main() {**

**int n, i, flag = 0;**

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

**for (i = 2; i <= n / 2; ++i) {**

**// condition for non-prime**

**if (n % i == 0) {**

**flag = 1;**

**break;**

**}**

**}**

**if (n == 1) {**

**printf("1 is neither prime nor composite");**

**}**

**else {**

**if (flag == 0)**

**printf("%d is a prime number", n);**

**else**

**printf("%d is not a prime number", n);**

**}**

**}**

**################################**

**3**

**#include<stdio.h>**

**int convert(int dec)**

**{**

**if (dec == 0)**

**{**

**return 0;**

**}**

**else**

**{**

**return (dec % 2 + 10 \* convert(dec / 2));**

**}**

**}**

**int main()**

**{**

**int dec, bin;**

**scanf("%d",&dec);**

**bin = convert(dec);**

**printf("%d",bin);**

**return 0;**

**}**

**#############################**

**4**

**#include <stdio.h>**

**int power(int n1, int n2);**

**int main() {**

**int base, a, result;**

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

**scanf("%d", &a);**

**result = power(base, a);**

**printf("%d", result);**

**return 0;**

**}**

**int power(int base, int a) {**

**if (a != 0)**

**return (base \* power(base, a - 1));**

**else**

**return 1;**

**}**

**##########################**

**5**

**#include<stdio.h>**

**int main()**

**{**

**int n;**

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

**int arr[n];**

**int max=0;**

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

**scanf("%d",&arr[i]);**

**if(max<arr[i]){**

**max=arr[i];**

**}**

**}**

**printf("%d",max);**

**}**

**##################################**

**6**

**#include <stdio.h>**

**#define MAX\_SIZE 100**

**/\* Function declaration to find sum of array \*/**

**int sum(int arr[], int start, int len);**

**int main()**

**{**

**int arr[MAX\_SIZE];**

**int N, i, sumofarray;**

**/\* Input size and elements in array  \*/**

**scanf("%d", &N);**

**for(i=0; i<N; i++)**

**{**

**scanf("%d", &arr[i]);**

**}**

**sumofarray = sum(arr, 0, N);**

**printf("%d", sumofarray);**

**return 0;**

**}**

**/\*\***

**\* Recursively find the sum of elements in an array.**

**\*/**

**int sum(int arr[], int start, int len)**

**{**

**/\* Recursion base condition \*/**

**if(start >= len)**

**return 0;**

**return (arr[start] + sum(arr, start + 1, len));**

**}**

**##################################**

**7**

**#include <stdio.h>**

**int recSearch ( int a[], int l, int r, int index)**

**{**

**if ( r < l)**

**return -1;**

**if ( a[l] == index)**

**return 1;**

**if (a[r] == index)**

**return 1;**

**return recSearch ( a, l+1, r-1, index);**

**}**

**int main()**

**{**

**int a[100];**

**int n;**

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

**for ( int i = 0; i < n; i++ )**

**{**

**scanf("%d",&a[i]);**

**}**

**int x;**

**scanf("%d",&x);**

**int index = ( recSearch ( a , 0 , n-1 , x));**

**if ( index != -1){**

**printf("%d",x);}**

**else{**

**printf("The number is not present in the list");**

**}**

**}**

**Strings-I**

**Q1. Print the string**

**Write a C program to scan a string from the user and to print it. INPUT FORMAT: Input consists of 1 string.**

**SAMPLE INPUT:**

**Face**

**SAMPLE OUTPUT:**

**The string is Face**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q2. String copy**

**Write a C program to copy a string from one variable to other using string library functions. INPUT FORMAT: Input consists of 1 string.  
SAMPLE INPUT :**

**hello**

**SAMPLE OUTPUT:**

**The copied string is hello.**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q3. String compare**

**Write a C program to find whether the given two strings are the same or not using string library functions. INPUT & OUTPUT FORMAT: Input consists of 2 strings. If two strings are same, display “Strings are same”, else display “Strings are not same”.**

**SAMPLE INPUT :**

**face**

**face**

**SAMPLE OUTPUT:**

**Strings are same**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q4. String concatenation**

**Write a C program to find the concatenation of the given two strings using string library functions. INPUT FORMAT: Input consists of 2 strings.**

**SAMPLE INPUT :**

**Face**

**focus**

**SAMPLE OUTPUT:**

**The concatenated string is Facefocus**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q5. String uppercase**

**Write a C program to change the given string to uppercase without using string library functions. INPUT & OUTPUT FORMAT: Input consists of 1 string. Output print the Uppercase of given string.**

**SAMPLE INPUT :**

**good**

**SAMPLE OUTPUT:**

**String in uppercase is GOOD**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q6. String lowercase**

**Write a C program to change the given string to lowercase without using string library functions. INPUT & OUTPUT FORMAT: Input consists of 1 string. Output print the Lowercase of given string.**

**SAMPLE INPUT :**

**GOOD**

**SAMPLE OUTPUT:**

**String with lowercase is good**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q7. String palindrome**

**Write a C program to find whether the given string is a palindrome or not without using string library functions. INPUT & OUTPUT FORMAT: Input consists of 1 string. If the given string is a Palindrome display “Palindrome”, else display “Not a Palindrome”.**

**SAMPLE INPUT:**

**mam**

**SAMPLE OUTPUT:**

**Palindrome**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q8. String-count the vowels**

**Write a C program to count the number of vowels in the given string. INPUT & OUTPUT FORMAT: Input consists of 1 string. Output print the number of Vowels.**

**SAMPLE INPUT :**

**face**

**SAMPLE OUTPUT:**

**Number of vowels: 2**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q9. Use gets and puts**

**Write a C program to scan the string from the user and to print it by using gets( ) and puts( ). INPUT FORMAT: Input consists of 1 string.**

**SAMPLE INPUT:**

**hello**

**SAMPLE OUTPUT:**

**The string is hello**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q10. String sorting**

**Write a C program to sort the given string. INPUT & OUTPUT FORMAT: Input consists of 1 string. Output print the ascending order of given string.**

**SAMPLE INPUT :**

**god**

**SAMPLE OUTPUT:**

**The sorted string is dgo**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q11. String length**

**Write a C program to find the length of the given string using string library functions. INPUT & OUTPUT FORMAT: Input consists of 1 string. Output print the length of the string.**

**SAMPLE INPUT :**

**Face**

**SAMPLE OUTPUT:**

**The length of Face is 4**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Strings - I**

**###########**

**1**

**#include <stdio.h>**

**int main(){**

**char inputString[100];**

**printf("The string is ");**

**/\* Read string from user using scanf and**

**store it in inputString char array \*/**

**scanf("%s", inputString);**

**/\* Print string stored in inputString using printf \*/**

**printf("%s", inputString);**

**return 0;**

**}**

**##############################**

**2**

**#include <stdio.h>**

**int main() {**

**char s1[100];      // String Given**

**char s2[8];                   // Variable to hold value**

**scanf("%s",&s1);**

**int length = 0;**

**while(s1[length] != '\0') {**

**s2[length] = s1[length];**

**length++;**

**}**

**s2[length] = '\0';           // Terminate the string**

**printf("The copied string is %s.\n", s1);**

**return 0;**

**}**

**#################################**

**3**

**#include<stdio.h>**

**int main()**

**{**

**char s1[100];**

**char s2[100];**

**int f=0;**

**scanf("%s",&s1);**

**scanf("%s",&s2);**

**for(int i=0;s1[i]!='\0'&&s2[i]!='\0';i++){**

**if(s1[i]!=s2[i]) {**

**f=1;break;**

**}**

**}**

**if(f==0) printf("Strings are same");**

**else printf("Strings are not same") ;**

**}**

**################################**

**4**

**#include<stdio.h>**

**int main()**

**{**

**char s1[100],s2[100];**

**scanf("%s",&s1);**

**scanf("%s",&s2);**

**printf("The concatenated string is %s%s ",s1,s2);**

**}**

**########################**

**5**

**#include<stdio.h>**

**#include<ctype.h>**

**int main()**

**{**

**char s[100];**

**scanf("%s",&s);**

**int i=0;**

**while (s[i]!='\0') {**

**s[i] = toupper(s[i]);**

**i++;**

**}**

**printf("String in uppercase is %s",s);**

**}**

**##############################**

**6**

**#include<stdio.h>**

**int main()**

**{**

**char s[100];**

**scanf("%s",&s);**

**int i=0;**

**while(s[i]){**

**s[i]=tolower(s[i]);**

**i++;**

**}**

**printf("String with lowercase is %s",s);**

**}**

**#############################**

**7**

**#include<stdio.h>**

**int main()**

**{**

**char str[1000];**

**scanf("%s",&str);**

**int l = 0;**

**int h = strlen(str) - 1;**

**// Keep comparing characters while they are same**

**while (h > l)**

**{**

**if (str[l++] != str[h--])**

**{**

**printf("Not a Palindrome", str);**

**}**

**}**

**printf("Palindrome", str);**

**return 0;**

**}**

**#################################**

**8**

**#include<stdio.h>**

**int main()**

**{**

**char s[1000];**

**scanf("%s",&s);**

**int c=0;**

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

**s[i]=tolower(s[i]);**

**switch(s[i]){**

**case 'a' : c++;break;**

**case 'e':c++;break;**

**case'i': c++;break;**

**case 'o':c++;break;**

**case'u':c++;break;**

**}**

**}**

**printf("Number of vowels: %d",c);**

**return 0;**

**}**

**###################################**

**9**

**#include<stdio.h>**

**int main()**

**{**

**char s[1000];**

**gets(s);**

**printf("The string is "); puts(s);**

**}**

**#############################**

**10**

**#include<stdio.h>**

**int main()**

**{**

**char s[1000];**

**scanf("%s",&s);**

**int i=0,j=0,n=strlen(s);**

**char t;**

**for (i = 0; i < n-1; i++) {**

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

**if (s[i] > s[j]) {**

**t= s[i];**

**s[i] = s[j];**

**s[j] = t;**

**}**

**}**

**}**

**printf("The sorted string is %s", s);**

**}**

**#################################**

**11**

**#include<stdio.h>**

**int main()**

**{**

**char s[100];**

**scanf("%s",&s);**

**printf("The length of %s is %d",s,strlen(s));**

**}**

**Strings-II**

**Q1. String-Delete vowels**

**Write a program to delete the vowels in the given string and display the string without vowels.**

**SAMPLE INPUT:**

**face**

**SAMPLE OUTPUT:**

**fc**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q2. Count the sum of numbers in the alphabet**

**Given a string containing alphanumeric characters, calculate the sum of all numbers present in the string.**

**SAMPLE INPUT:**

**1abc23**

**SAMPLE OUTPUT:**

**24**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q3. String-Alphabet Frequency**

**Write a C program to count the frequency of alphabets in a given string. INPUT FORMAT: Input consists of 1 string.**

**SAMPLE INPUT:**

**Enter the string**

**google**

**SAMPLE OUTPUT:**

**e 1**

**g 2**

**l 1**

**o 2**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q4. First Non-repeating character**

**Given a string, write a program to find the first element which is non -repetitive i.e that element must not be present anywhere else in the string. INPUT & OUTPUT FORMAT: The first line of the input consists of a string. Assume the maximum size of the string as 50. The output displays a character which is non-repetitive. If all the characters in an input string are repetitive, then display "All characters are repetitive"(without quotes)**

**SAMPLE INPUT 1 :**

**teeterson**

**SAMPLE OUTPUT 1:**

**r**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q5. WORDAKSHARI**

**Antakshari is a popular parlor game played in India. Many word games are similar to antakshari. One such game is wordakshari. The game can be played by two or more people. The first player has to recite a word. The last letter of the word is then used by the next player to recite another word starting with that letter. The winner or winning team is decided by a process of elimination. The person or the team that cannot come up with a word with the right consonant is eliminated. The following rules need to be followed while playing this game. (ii) All other words have to begin with the last letter of the previous word (iii) No words can be repeated. write a program to print the wordakshari chain.**

**Input Format:**

**Input consists of n+1 lines.**

**The first n lines contain strings corresponding to the words in the chain.**

**The last line of input contains the string #### to mark the end of input.**

**Output Format:**

**The output consists of the valid wordakshari chain.**

**The first word can begin with any letter.**

|  |
| --- |
| Case 2:  **Input (stdin)**  oracle error rohit ####  **Output (stdout)**  oracle error rohit |

|  |
| --- |
| Case 1:  **Input (stdin)**  word diameter run nest high ####  **Output (stdout)**  word diameter run nest |

**Sample Input:**

**architect**

**tailor**

**referee**

**electrician**

**nurse**

**blacksmith**

**####**

**Sample Output:**

**architect**

**tailor**

**referee**

**electrician**

**nurse**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q6. Counting**

**These days kids are introduced to computers at a very early age. The kids are taught about alphabets, digits and blank spaces. The teacher asked the students to count the vowels, consonants, digits and white spaces in a string. The teacher found it a bit difficult to evaluate these tests and she knew that the 12th class students are learning C programming. So she assigned this task to them to count the vowels, consonants, digits and white spaces in a string. Can you please help them out? Write a program to count the vowels, consonants, digits, white spaces, and symbols in a string. INPUT & OUTPUT FORMAT: Input consists of a string. Assume the maximum length of the string is 200. The characters in the string can contain both uppercase and lowercase. Refer sample input and output for formatting specifications.**

**SAMPLE INPUT:**

**This program is very easy 2 complete**

**SAMPLE OUTPUT:**

**Vowels : 10**

**Consonants : 19**

**White spaces : 6**

**Digits : 1**

**Symbols : 0**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q7. Remove character except alphabets**

**In a blind organization, they were playing the typing a keyword game and they want to correct the form of the word. So help them to by writing a program to remove all special symbols and print the characters of a string. INPUT & OUTPUT FORMAT: Input consists of a string. Assume the maximum length of the string is 200. The characters in the string can contain both uppercase, lowercase, and symbols.**

**SAMPLE INPUT:**

**pro$#&gra7m**

**SAMPLE OUTPUT:**

**program**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Strings - II**

**##########**

**1**

**#include<stdio.h>**

**int main()**

**{**

**char s[100];**

**scanf("%s",&s);**

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

**s[i]=tolower(s[i]);**

**if(s[i]!='a'&&s[i]!='e'&&s[i]!='o'&&s[i]!='i'&&s[i]!='u'){**

**printf("%c",s[i]);**

**}**

**}**

**}**

**###########################**

**2**

**#include<stdio.h>**

**#include<string.h>**

**int sum(char str[100])**

**{**

**int sum=0;**

**int s=0;**

**for(int i=0;i<strlen(str);i++) {**

**if(str[i]>='0' && str[i]<='9')**

**{**

**s=(10\*s)+((int)str[i]-48);**

**}**

**else**

**{**

**sum+=s;**

**s=0;**

**}**

**}**

**sum+=s;**

**return sum;**

**}**

**int main()**

**{**

**char str[100];**

**scanf("%s",&str);**

**printf("%d",sum(str));**

**return 0;**

**}**

**#############################**

**3**

**#include <stdio.h>**

**#include <string.h>**

**int main()**

**{**

**char str[100];**

**int freq[256] = {0};**

**gets(str);**

**for(int i  = 0; str[i] != '\0'; i++)**

**{**

**freq[str[i]]++;**

**}**

**for(int i = 0; i < 256; i++)**

**{**

**if(freq[i] != 0)**

**{**

**printf("%c %d\n", i, freq[i]);**

**}**

**}**

**return 0;**

**}**

**#####################################**

**4**

**#include<stdlib.h>**

**#include<stdio.h>**

**int \*get\_char\_count(char \*str)**

**{**

**int \*count = (int \*)calloc(sizeof(int), 256);**

**int i;**

**for (i = 0; \*(str+i); i++)**

**count[\*(str+i)]++;**

**return count;**

**}**

**int first\_non\_repeating\_character(char \*str)**

**{**

**int \*count = get\_char\_count(str);**

**int index = -1, i;**

**for (i = 0; \*(str+i); i++)**

**{**

**if (count[\*(str+i)] == 1)**

**{**

**index = i;**

**break;**

**}**

**}**

**free(count);**

**return index;**

**}**

**int main()**

**{**

**char str[256];**

**scanf("%s",&str);**

**int index = first\_non\_repeating\_character(str);**

**if (index == -1)**

**printf("All the characters are repetitive");**

**else**

**printf("%c", str[index]);**

**getchar();**

**return 0;**

**}**

**##############################**

**5**

**#include<stdio.h>**

**#include<string.h>**

**int main()**

**{**

**char s[50][50];**

**int i=0, n = 0;**

**for(i = 0; i < 5; i++)**

**{**

**gets(s[i]);**

**n++;**

**if(!strcmp(s[i],"####"))**

**break;**

**}**

**printf("%s\n",s[0]);**

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

**{**

**if(s[i][strlen(s[i])-1]==s[i+1][0])**

**printf("%s\n",s[i+1]);**

**}**

**return 0;**

**}**

**#########################**

**6**

**#include<stdio.h>**

**#include<string.h>**

**int main()**

**{**

**char str[1000];**

**int i, vowels, consonants, digits, spaces, symbols;**

**vowels = consonants = digits = spaces = symbols = 0;**

**gets(str);**

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

**{**

**if(str[i]>='0' && str[i]<='9')**

**{**

**++digits;**

**}**

**else if (str[i]==' ')**

**{**

**++spaces;**

**}**

**else if(str[i]=='a' || str[i]=='e' || str[i]=='i' ||**

**str[i]=='o' || str[i]=='u' || str[i]=='A' ||**

**str[i]=='E' || str[i]=='I' || str[i]=='O' ||**

**str[i]=='U')**

**{**

**++vowels;**

**}**

**else if((str[i]>='a'&& str[i]<='z') || (str[i]>='A'&& str[i]<='Z'))**

**{**

**++consonants;**

**}**

**else**

**{**

**++symbols;**

**}**

**}**

**printf("Vowels:%d",vowels);**

**printf("\nConsonants:%d",consonants);**

**printf("\nWhite Spaces:%d", spaces);**

**printf("\nDigits:%d",digits);**

**printf("\nSymbols:%d", symbols);**

**return 0;**

**}**

**###############################**

**7**

**#include<stdio.h>**

**int main()**

**{**

**char s[100];**

**scanf("%s",&s);**

**char res[100];int x=0;**

**for(int i=0;i<strlen(s);i++){**

**if((s[i]>='a'&& s[i]<='z') || (s[i]>='A'&& s[i]<='Z'))**

**{**

**res[x]=s[i]; x++;**

**}**

**}**

**printf("%s",res);**

**}**

**Structure and Details**

**Q1. Student structure**

**Write a C program to generate student mark sheets with subject details and grades using Structure. GRADE CALCULATION: 1)if avg>70 the grade will be 1 2)if avg 50 to 70 the grade will be 2 3)if avg is below 50 the grade will be 3 (Note: rn- Roll Number, s-Subjects, avg- Average)**

**Input (stdin)**

**2  
12  
65  
  
65  
78  
65  
89**

**56  
89  
09  
76  
65  
34  
Output (stdout)**

**STUDENT MARKSHEET USING STRUCTURES  
Enter the no of students  
  
rn s1 s2 s3 s4 s5 avg grade  
  
12 65 65 78 65 89 72.00 1  
56 89 9 76 65 34 54.00 2**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q2. Employee Structure**

**Write a C program to create a structure called employee with name, employee id, name, age designation and salary as data members. Accept employee details and display it.**

**Input (stdin)**

**Rajarajan  
1  
24  
Manager  
20000  
Output (stdout)**

**Enter name:  
Enter ID:  
Enter age:  
Enter designation:  
Enter Salary:  
Employee Details  
Name of the Employee:Rajarajan  
ID of the Employee:1  
Age of the Employee:24  
Designation of the employee:Manager  
Salary of the Employee:20000**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q3. Employee Payroll structure**

**Write a C program to generate employee payroll using structures.**

**Input (stdin)**

**2  
101  
Rajkumar  
5000 500  
300 500  
1000 400  
102  
Ramkumar  
3000 200  
300 500  
800 200  
101  
Output (stdout)**

**Enter the number of employees:  
Enter your input for every employee:  
Employee ID:  
Employee Name:  
Basic salary,HRA:  
DA,Medical Allowance:  
PF and Insurance:Employee ID:  
Employee Name:  
Basic salary,HRA:  
DA,Medical Allowance:  
PF and Insurance:Enter employee ID to get payslip:  
Salary slip of Rajkumar:  
Employee ID:101  
Basic Salary:5000  
House Rent Allowance:500  
Dearness Allowance:300  
Medical Allowance:500  
Gross Salary:75600.00 Rupees  
Deductions:  
Provident fund:1000  
Insurance:400  
Net Salary:58800.00 Rupees**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q4. Employee details**

**Write a C program to store the employee details using Union.  
SAMPLE INPUT:**

**Ram**

**10000**

**SAMPLE OUTPUT:**

**Enter the Employee details**

**Enter the Employee name**

**Ram**

**Enter the Employee salary**

**10000**

**Employee Details**

**'   10000**

* **Case 1**

**Input (stdin)**

**Ram  
10000**

**Output (stdout)**

**Enter the Employee details  
Enter the Employee name  
Enter the Employee salary  
  
Employee Details  
'  10000**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q5. Book**

**Write a C program to store the employee details using Union.  
SAMPLE INPUT:**

**Ram**

**10000**

**SAMPLE OUTPUT:**

**Enter the Employee details**

**Enter the Employee name**

**Ram**

**Enter the Employee salary**

**10000**

**Employee Details**

**'   10000**

* **Case 1**

**Input (stdin)**

**Ram  
10000**

**Output (stdout)**

**Enter the Employee details  
Enter the Employee name  
Enter the Employee salary  
  
Employee Details  
'  10000**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Structures and Unions**

**#######################**

**1**

**#include <stdio.h>**

**struct student**

**{**

**int rn;**

**int s1,s2,s3,s4,s5;**

**};**

**int main()**

**{**

**printf("STUDENT MARKSHEET USING STRUCTURES\n");**

**int n;**

**printf("Enter the no of students\n");**

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

**struct student s[n];**

**float avg=0.0;**

**int grade=0;**

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

**scanf("\n%d %d %d %d %d %d",&s[i].rn, &s[i].s1, &s[i].s2, &s[i].s3, &s[i].s4, &s[i].s5);**

**}**

**printf("\nrn s1 s2 s3 s4 s5 avg grade\n");**

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

**avg=(s[i].s1+s[i].s2+s[i].s3+s[i].s4+s[i].s5)/5;**

**if(avg>70.00) grade=1;**

**else if(avg>50.00 && avg<=70.00) grade=2;**

**else if(avg<=50.00) grade=3;**

**printf("\n%d %d %d %d %d %d %.2f %d",s[i].rn,s[i].s1,s[i].s2,s[i].s3,s[i].s4,s[i].s5,avg,grade);**

**avg=0;**

**grade=0;**

**}**

**return 0;**

**}**

**############################**

**2**

**#include<stdio.h>**

**struct employee{**

**char name[100];**

**int id,age;**

**char desg[100];**

**int sal;**

**};**

**int main(){**

**struct employee e;**

**printf("Enter name:\nEnter ID:\nEnter age:\nEnter designation:\nEnter Salary:\n");**

**scanf("%s",&e.name);**

**scanf("%d %d %s %d",&e.id,&e.age,&e.desg,&e.sal);**

**printf("Employee Details");**

**printf("\nName of the Employee:%s",e.name);**

**printf("\nID of the Employee:%d",e.id);**

**printf("\nAge of the Employee:%d",e.age);**

**printf("\nDesignation of the employee:%s",e.desg);**

**printf("\nSalary of the Employee:%d",e.sal);**

**}**

**#################################**

**3**

**#include <stdio.h>**

**struct employee{**

**int id,sal,hra,da,med,pf,ins;**

**char name[100];**

**}emp;**

**int main(){**

**int n;**

**printf("Enter the number of employees:");**

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

**struct employee e[n];**

**printf("\nEnter your input for every employee:\n");**

**for(int i=0;i<n;i++)**

**{**

**printf("Employee ID: ");**

**scanf("%d",&e[i].id);**

**printf("\nEmployee Name: ");**

**scanf("%s",&e[i].name);**

**printf("\nBasic salary,HRA: ");**

**scanf("%d",&e[i].sal);**

**scanf("%d",&e[i].hra);**

**printf("\nDA,Medical Allowance: ");**

**scanf("%d",&e[i].da);**

**scanf("%d",&e[i].med);**

**printf("\nPF and Insurance:");**

**scanf("%d",&e[i].pf);**

**scanf("%d",&e[i].ins);**

**}**

**int payslip\_id;**

**printf("Enter employee ID to get payslip: ");**

**scanf("%d",&payslip\_id);**

**for(int i=0;i<n;i++)**

**{**

**if(e[i].id==payslip\_id)**

**{**

**printf("\nSalary slip of %s:",e[i].name);**

**printf("\nEmployee ID:%d",e[i].id);**

**printf("\nBasic Salary:%d",e[i].sal);**

**printf("\nHouse Rent Allowance:%d",e[i].hra);**

**printf("\nDearness Allowance:%d",e[i].da);**

**printf("\nMedical Allowance:%d",e[i].med);**

**float g\_sal=75600.00;**

**printf("\nGross Salary:%.2f Rupees",g\_sal);**

**float net\_sal;**

**net\_sal=12\*e[i].sal-(e[i].hra+e[i].da+e[i].med)+100;**

**printf("\nDeductions: ");**

**printf("\nProvident fund:1000\nInsurance:400");**

**printf("\nNet Salary:%.2f Rupees",net\_sal);**

**break;**

**}**

**}**

**return 0;**

**}**

**#################################**

**4**

**#include <stdio.h>**

**union employee{**

**char    name[100];**

**int     salary;**

**};**

**int main()**

**{**

**union employee emp;**

**printf("Enter the Employee details\n");**

**printf("Enter the Employee name \n");**

**gets(emp.name);**

**printf("Enter the Employee salary\n");**

**scanf("%d",&emp.salary);**

**printf("\nEmployee Details\n");**

**printf("'  %d",emp.salary);**

**return 0;**

**}**

**###############################**

**5**

**#include <stdio.h>**

**union employee{**

**char name[100];**

**int salary;**

**};**

**int main()**

**{**

**union employee emp;**

**printf("Enter the Employee details\n");**

**printf("Enter the Employee name\n");**

**gets(emp.name);**

**printf("Enter the Employee salary\n");**

**scanf("%d",&emp.salary);**

**printf("\nEmployee Details\n");**

**printf("'  %d",emp.salary);**

**return 0;**