

**Date:****Ex. No.:****Day Old Bread****Problem Statement:**

A bakery sells loaves of bread for \$3.49 each. Day old bread is discounted by 60 percent. Write a program that begins by reading the number of loaves of day-old bread being purchased from the user. Then your program should display the regular price for the bread, the discount because it is a day old, and the total price. Each of these amounts should be displayed on its own line with an appropriate label. All of the values should be displayed using two decimal places.

**Input Format**

Read the number of day old loaves.

**Output Format**

First line, print Regular price: price Second line, print Discount: discount Third line, print Total: total

**Note:** All of the values should be displayed using two decimal places.

**Sample Input 1**

10

**Sample Output 1**

Regular price: 34.90 Discount: 20.94 Total: 13.96

**Program:**

```
#include <stdio.h>

int main()
{
    int a;
    scanf ("%d", &a);
    printf ("Regular price: %.2f\n Discount: %.2f\n"
           "In Total: %.2f\n", a * 3.49, a * 3.49 * 0.6,
                  a * 3.49 * 0.4);

    return 0;
}
```

↓

**Ex. No. :****Date :**

### **Goki and his Breakup**

**Problem Statement:**

Goki recently had a breakup, so he wants to have some more friends in his life. Goki has  $N$  people who he can be friends with, so he decides to choose among them according to their skills set  $Y_i (1 \leq i \leq n)$ . He wants atleast  $X$  skills in his friends.  
Help Goki find his friends.

**Input Format**

First line contains a single integer  $X$  - denoting the minimum skill required to be Goki's friend. Next line contains one integer  $Y$  - denoting the skill of the person.

**Output Format**

Print if he can be friend with Goki. 'YES' (without quotes) if he can be friends with Goki else 'NO' (without quotes).

**Constraints**

$1 \leq N \leq 1000000$   $1 \leq X, Y \leq 1000000$

**SAMPLE INPUT 1**

100  
110

**SAMPLE OUTPUT 1**

YES

**Program:**

```
#include <stdio.h>

int main()
{
    int x,y;
    scanf("%d %d", &x, &y);
    if (y >= x)
    {
        printf("YES");
    }
    else
    {
        printf("NO");
    }
}
```

Ex. No. :

Date :

### Say no to Handshakes!!!

#### Problem Statement:

Before the outbreak of corona virus to the world, a meeting happened in a room in Wuhan. A person who attended that meeting had COVID-19 and no one in the room knew about it! So, everyone started shaking hands with everyone else in the room as a gesture of respect and after meeting unfortunately everyone got infected! Given the fact that any two persons shake hand exactly once, can you tell the total count of handshakes happened in that meeting?

**Say no to shakehands. Regularly wash your hands. Stay Safe.**

#### Input Format

Read an integer N, the total number of people attended that meeting.

#### Output Format

Print the number of handshakes.

#### Constraints

$0 < N < 10^6$

#### SAMPLE INPUT 1

1

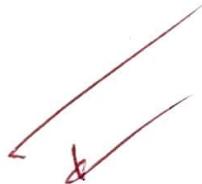
#### SAMPLE OUTPUT

0

**Program:**

```
#include<stdio.h>

int main()
{
    int a,b;
    scanf("%d", &a);
    b = a * (a-1) / 2;
    printf("%d", b);
}
```



b

Ex. No. :

Date :

**Back to School****Problem Statement:**

In our school days, all of us have enjoyed the Games period. Raghav loves to play cricket and is Captain of his team. He always wanted to win all cricket matches. But only one last Games period is left in school now. After that he will pass out from school.

So, this match is very important to him. He does not want to lose it. So he has done a lot of planning to make sure his teams wins. He is worried about only one opponent - Jatin, who is very good batsman.

Raghav has figured out 3 types of bowling techniques, that could be most beneficial for dismissing Jatin. He has given points to each of the 3 techniques.

You need to tell him which is the maximum point value, so that Raghav can select best technique.

3 numbers are given in input. Output the maximum of these numbers.

**Input Format:**

Three space separated integers.

**Output Format:**

Maximum integer value

**SAMPLE INPUT**

8 6 1

**SAMPLE OUTPUT**

8

**Program:**

```
#include <stdio.h>

int main()
{
    int a, b, c;

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

    int x = a;

    if (b > x)
    {
        x = b;
    }

    if (c > x)
    {
        x = c;
    }

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

Ex. No. :

Date :

### Same Digit

**Problem Statement:**

Write a program to read two integer values and print true if both the numbers end with the same digit, otherwise print false.

Example: If 698 and 768 are given, program should print true as they both end with 8.

**Sample Input 1**

25 53

**Sample Output 1**

false

**Sample Input 2**

27 77

**Sample Output 2**

true

**Program:**

```
#include <stdio.h>

int main()
{
    int x,y;
    scanf("%d%d", &x, &y);
    if (x==y) {
        printf("true");
    }
    else {
        printf("false");
    }
}
```

d

Ex. No. :

Date :

### Intro to Conditional Statements

**Problem Statement:**

In this challenge, we're getting started with conditional statements.

**Task**

Given an integer,  $n$ , perform the following conditional actions:

- If  $n$  is odd, print **Weird**
- If  $n$  is even and in the inclusive range of **2** to **5**, print **Not Weird**
- If  $n$  is even and in the inclusive range of **6** to **20**, print **Weird**
- If  $n$  is even and greater than **20**, print **Not Weird**

Complete the stub code provided in your editor to print whether or not  $n$  is weird.

**Input Format**

A single line containing a positive integer,  $n$ .

**Constraints**

- $1 < n < 100$

**Output Format**

Print **Weird** if the number is weird; otherwise, print **Not Weird**.

**Sample Input 0**

3

**Sample Output 0**

Weird

**Program:**

```
#include <stdio.h>

int main()
{
    int n;
    scanf("%d", &n);
    if (n % 2 == 0) {
        if (n >= 2 && n <= 5) {
            printf("Not weird");
        }
        if (n >= 6 && n <= 20) {
            printf("Weird");
        }
        if (n > 20) {
            printf("Not Weird");
        }
    }
    else {
        printf("Weird");
    }
}
```

~~2~~

**Ex. No. :****Date :**

### Pythagorean Triples

**Problem Statement:**

Three numbers form a Pythagorean triple if the sum of squares of two numbers is equal to the square of the third.

For example, 3, 5 and 4 form a Pythagorean triple, since  $3*3 + 4*4 = 25 = 5*5$

You are given three integers, a, b, and c. They need not be given in increasing order. If they form a Pythagorean triple, then print "yes", otherwise, print "no". Please note that the output message is in small letters.

**Sample Input 1**

3  
5  
4

**Sample Output 1**

yes

**Program:**

```
#include <stdio.h>

int main() {
    int a, b, c;
    scanf ("%d %d %d", &a, &b, &c);
    if (a*a+b*b == c*c) {
        printf ("yes");
    }
    else if (a*a+c*c == b*b) {
        printf ("yes");
    }
    else if (b*b+c*c == a*a) {
        printf ("yes");
    }
    else {
        printf ("no");
    }
    return 0;
}
```

✓

**Ex. No. :****Date :****Name That Shape****Problem Statement:**

Write a program that determines the name of a shape from its number of sides. Read the number of sides from the user and then report the appropriate name as part of a meaningful message. Your program should support shapes with anywhere from 3 up to (and including) 10 sides. If a number of sides outside of this range is entered then your program should display an appropriate error message.

**Sample Input 1**

3

**Sample Output 1**

Triangle

**Sample Input 2**

7

**Sample Output 2**

Heptagon

**Sample Input 3**

11

**Sample Output 3**

The number of sides is not supported.

**Program:**

```
#include<stdio.h>
int main() {
    int n;
    scanf("%d", &n);
    if (n == 3) {
        printf("Triangle");
    }
    else if (n == 4) {
        printf("square");
    }
    else if (n == 5) {
        printf("Pentagon");
    }
    else if (n == 6) {
        printf("Hexagon");
    }
    else if (n == 7) {
        printf("Heptagon");
    }
    else if (n == 8) {
        printf("Octagon");
    }
    else if (n == 9) {
        printf("Nonagon");
    }
    else if (n == 10) {
        printf("Decagon");
    }
}
```

```
else
{
    printf("The number of sides is not supported.");
}
}
```

Ex. No. :

Date :

**Chinese Zodiac****Problem Statement:**

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

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

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

**Sample Input 1**

2004

**Sample Output 1**

Monkey

**Sample Input 2**

2010

**Sample Output 2**

Tiger

**Program:**

```
#include <stdio.h>

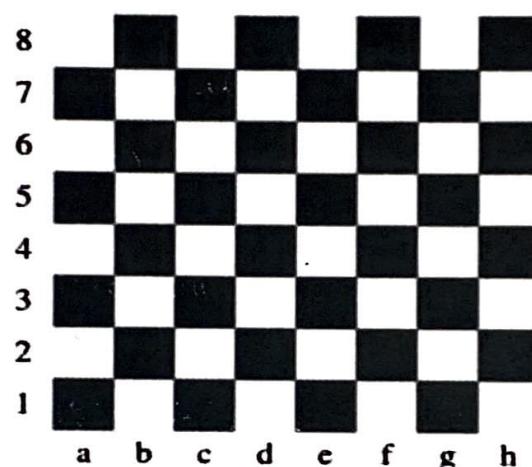
int main()
{
    int year;
    scanf("%d", &year);
    if (year % 12 == 8)
    {
        printf("Dragon");
    }
    else if (year % 12 == 9)
    {
        printf("Snake");
    }
    else if (year % 12 == 10)
    {
        printf("Horse");
    }
    else if (year % 12 == 11)
    {
        printf("Sheep");
    }
    else if (year % 12 == 0)
    {
        printf("Monkey");
    }
    else if (year % 12 == 1)
    {
        printf("Rooster");
    }
}
```

```
else if (year % 12 == 2) {  
    printf("Dog");  
}  
  
else if (year % 12 == 3) {  
    printf("Pig");}  
  
else if (year % 12 == 4) {  
    printf("Rat");}  
  
else if (year % 12 == 5) {  
    printf("Ox");}  
  
else if (year % 12 == 6) {  
    printf("Tiger");}  
}  
  
else {  
    printf("Hare");}  
}
```



**Ex. No. :****Date :****What Color Is That Square?****Problem Statement:**

Positions on a chess board are identified by a letter and a number. The letter identifies the column, while the number identifies the row, as shown below:



Write a program that reads a position from the user. Use an if statement to determine if the column begins with a black square or a white square. Then use modular arithmetic to report the color of the square in that row. For example, if the user enters a1 then your program should report that the square is black. If the user enters d5 then your program should report that the square is white. Your program may assume that a valid position will always be entered. It does not need to perform any error checking.

**Sample Input 1**

a 1

**Sample Output 1**

The square is black.

**Sample Input 2**

d 5

**Sample Output 2**

The square is white.

**Program:**

```
#include<stdio.h>

int main()
{
    int num, sum;
    char alpha;
    scanf ("%c%d", &alpha, &num);
    sum = alpha + num;
    if (sum % 2 == 0)
    {
        printf ("The square is black");
    }
    else
    {
        printf ("The square is white");
    }
}
```

d

**Ex. No. :****Date :****Day of Year****Problem Statement:**

Some data sets specify dates using the year and day of year rather than the year, month, and day of month. The day of year (DOY) is the sequential day number starting with day 1 on January 1st.

There are two calendars - one for normal years with 365 days, and one for leap years with 366 days. Leap years are divisible by 4. Centuries, like 1900, are not leap years unless they are divisible by 400. So, 2000 was a leap year.

To find the day of year number for a standard date, scan down the Jan column to find the day of month, then scan across to the appropriate month column and read the day of year number. Reverse the process to find the standard date for a given day of year.  
Write a program to print the Day of Year of a given date, month and year.

**Sample Input 1**

18  
6  
2020

**Sample Output 1**

170

**Program:**

```
#include <stdio.h>

int main() {
    int d, m, y, feb;
    scanf ("%d.%d.%d", &d, &m, &y);
    if ((y % 100 == 0 || y % 400 == 0) || (y % 4 == 0))
        feb = 29;
    else
        feb = 28;
    switch (m) {
        case 1:
            printf ("%d", d);
            break;
        case 2:
            printf ("%d", 31 + d);
            break;
        case 3:
            printf ("%d", 31 + feb + d);
            break;
        case 4:
    }
}
```

```
printf ("%d", 31 + feb + 31 + d);  
break;
```

Case 5:

```
printf ("%d", 31 + feb + 31 + 30 + d);  
break;
```

Case 6:

```
printf ("%d", 31 + feb + 31 + 30 + 31 + d);  
break;
```

Case 7:

```
printf ("%d", 31 + feb + 31 + 30 + 31 + 30 + d);  
break;
```

Case 8:

```
printf ("%d", 31 + feb + 31 + 30 + 31 + 30 + 31 + d);  
break;
```

Case 9:

```
printf ("%d", 31 + feb + 31 + 30 + 31 + 30 + 31 + 30 + d);  
break;
```

Case 10:

```
printf ("%d", 31 + feb + 31 + 30 + 31 + 30 + 31 + 30 + 31 + d);  
break;
```

Case 11:

```
printf ("%d", 31 + feb + 31 + 30 + 31 + 30 + 31 + 30 + 31 + d);  
break;
```

Case 12:

```
printf ("%d", 31 + feb + 31 + 30 + 31 + 30 + 31 + 30 + 31 + 30 + d);  
break;
```