Ex. No.: **Date:**

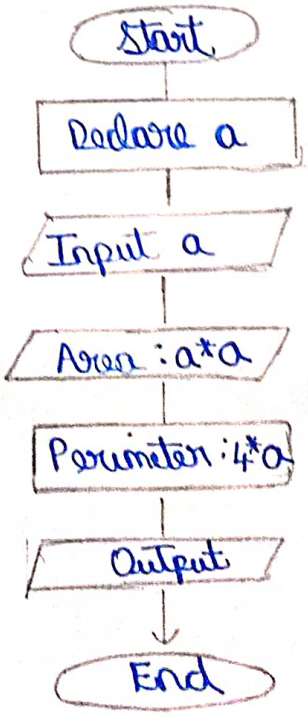
Calculate Area and Perimeter

Write an Algorithm and draw a Flowchart to Calculate the area and perimeter of a square.

Algorithm:

Step 1 : Start
Step 2 : Declare variable a
Step 3 : Input a
Step 4 : Area : a*a
Step 5 : Perimeter : 4*a
Step 6 : End

Flowchart:



Sample Output

a = 2
Area = 4
Perimeter = 8

Ex. No.: **Date:**

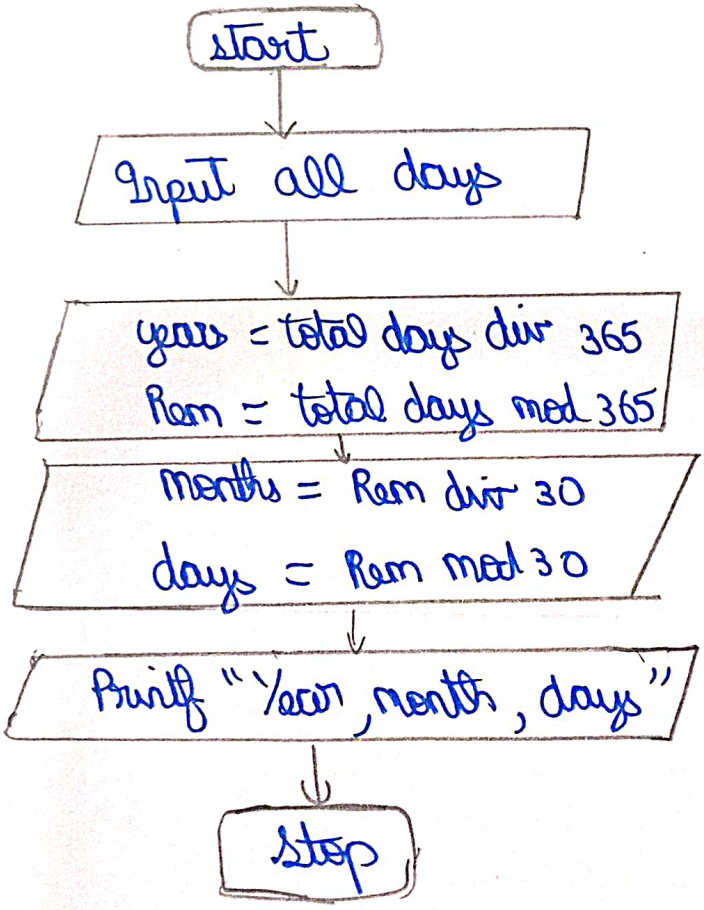
Days to Year Conversion

Write an Algorithm and draw a Flowchart to convert the given days into years & months.

Algorithm:

Step 1 : Start
Step 2 : [Input the number of days] Input total days.
step 3: [Compute years] years = total days div 365.
Step 4 : [Compute remaining days] Rem = total days mod 365.
Step 5 : [Compute months] months = Rem div 30
Step 6 : [Compute remaining days] days = Rem mod 30
Step 7 : Print "years, months, days".
Step 8 : stop.

Flowchart:



Ex. No.: **Date:**

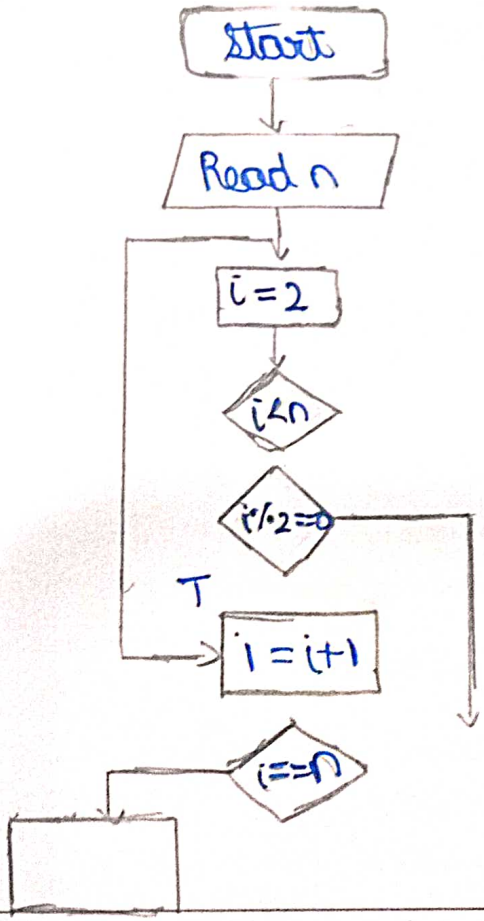
Prime Number

Write an Algorithm and draw a Flowchart to check whether the given number is Prime or not.

Algorithm:

Step 1 : start
Step 2 : Read n, step 3 : initialize i = 2
Step 4 : if i = 2 and i < n . go to step 8.
Step 5 : if i % 2 == 0 . go to step 8.
Step 6 : i == n
step 7 : i = i+1, go to step 4
step 8 : if i == n, printf "not prime"
else print "prime".
Step 9 : stop

Flowchart:



Ex. No.: **Date:**

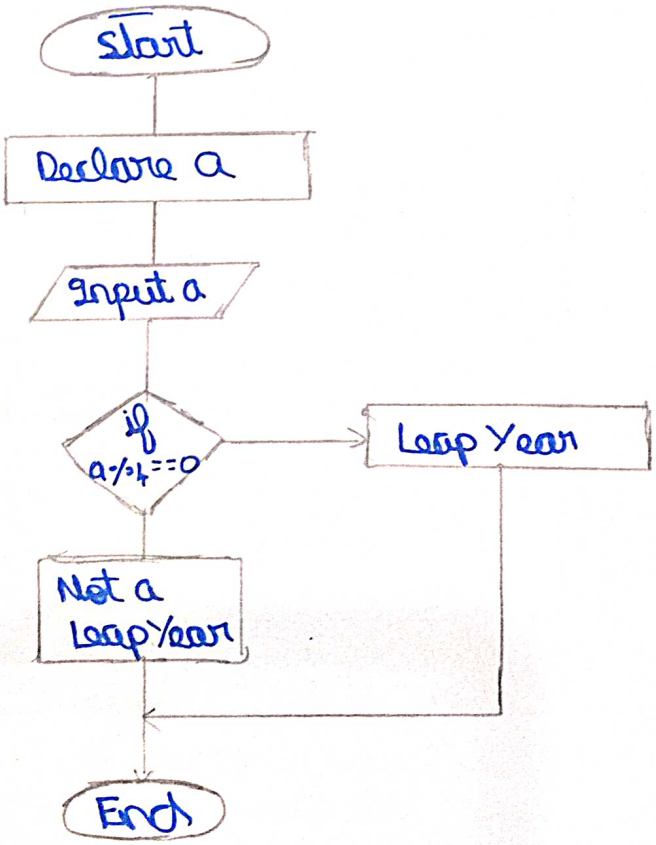
Leap Year

Write an Algorithm and draw a Flowchart to check whether the given year is Leap year or not.

Algorithm:

Step 1 : Start
Step 2 : Declare a
step 3: Input a
step 4 : if (a %4== 0) (Leap year)
step 5 : else (Not Leap year)
step 6 : output
step 7 : End

Flowchart:

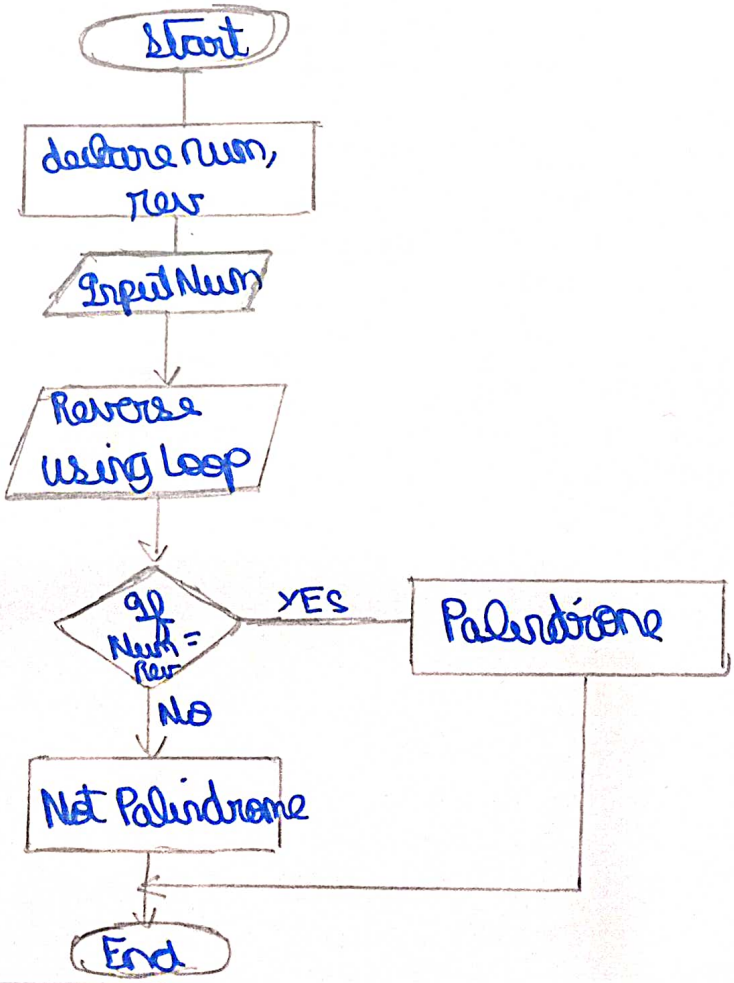


Write an Algorithm and draw a Flowchart to check whether the given number is palindrome number or not.

Algorithm:

step 1 : Start
step 2 : Declare num, rev
step 3 : Input num
step 4 : reverse using for loop
step 5 : if num = rev
step 6 : else
step 7 : output
step 8 : End

Flowchart:



Ex. No.: **Date:**

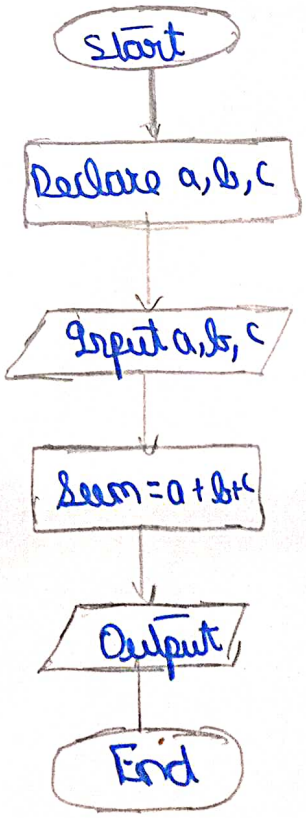
Sum of Digits

Write an Algorithm and draw a Flowchart to calculate the sum of digits in the given number.

Algorithm:

Step 1 : Start
Step 2 : Declare a, b, c
Step 3 : Input a, b, c
Step 4 : Sum = a + b + c
Step 5 : Declare Output
Step 6 : End

Flowchart:



Overview of C, Constants, Variables and Data Types

Ex. No.: **Date:**

Say "Hello, World!" With C

Problem Statement:

This is a simple challenge to help you practice printing to stdout.
We're starting out by printing the most famous computing phrase of all time! In the editor below, use either printf or cout to print the string Hello, World! to stdout.

Input Format
You do not need to read any input in this challenge.

Output Format
Print ***Hello, World!*** to stdout.

Sample Output 1
Hello, World!

Program:

```
#include <stdio.h>
int main()
{
    printf("Hello, World!")
    return 0;
}
```

Ex. No.: Date:

Playing with Characters

Problem Statement:

This challenge will help you to learn how to take a character, a string and a sentence as input in C. To take a single character **ch** as input, you can use scanf("%c", &ch); and printf("%c", ch) writes a character specified by the argument char to stdout:

```
char ch;
scanf("%c", &ch);
printf("%c", ch);
```

This piece of code prints the character **ch**. You can take a string as input in C using scanf("%s", s). But it accepts string only until it finds the first space.
In order to take a line as input, you can use scanf("%[^\n] %*c", s); where s is defined as chars [**MAX_LEN**] where MAX_LEN is the maximum size of s. Here, [] is the scanset character. ^\n stands for taking input until a newline isn't encountered. Then, with this %*c, it reads the newline character and here, the used * indicates that this newline character is discarded.
Note: After inputting the character and the string, inputting the sentence by the above mentioned statement won't work. This is because, at the end of each line, a new line character(\n) is present. So, the statement: scanf("%[^\n] %*c", s); will not work because the last statement will read a newline character from the previous line. This can be handled in a variety of ways and one of them being: scanf("\n"); before the last statement.
Task: You have to print the character, **ch**, in the first line. Then print s in next line. In the last line print the sentence, **sen.**

Input Format
First, take a character, **ch** as input. Then take the string, **s** as input. Lastly, take the sentence **sen** as input

Output Format
Print three lines of output. The first line prints the character, **ch.** The second line prints the string, **s.** The third line prints the sentence, **sen.**

Sample Input 1
C
program
Programming using C
Sample Output 1
C
program
Programming using C

Program:

```
#include <stdio.h>
int main()
{
    char ch, sen[100], s, s[100];
    scanf ("%c", "\n", &ch);
    scanf ("%s", \n", &s);
    getchar ();
    scanf ("%[^\n]%.*c", sen);
    printf ("%c\n", ch);
    printf ("%s\n", s);
    printf ("%s\n", sen);
    return 0;
}
```

Ex. No.: **Date:**

Sum and Difference of Two Numbers

Problem Statement:

The fundamental data types in c are int, float and char. Today, we're discussing int and float data types.

The printf() function prints the given statement to the console. The syntax is printf("format string",argument_list);. In the function, if we are using an integer, character, string or float as argument, then in the format string we have to write %d (integer), %c (character), %s (string), %f (float) respectively.

The scanf() function reads the input data from the console. The syntax is scanf("format string",argument_list);. For ex: The scanf("%d",&number) statement reads integer number from the console and stores the given value in variable ***number***.
To input two integers separated by a space on a single line, the command is scanf("%d %d", &n, &m), where ***n*** and ***m*** are the two integers.

Task
Your task is to take two numbers of int data type, two numbers of float data type as input and output their sum:

1. Declare ***4*** variables: two of type int and two of type float.
2. Read ***2*** lines of input from stdin (according to the sequence given in the 'Input Format' section below) and initialize your ***4*** variables.
3. Use the + and - operator to perform the following operations:
   - Print the sum and difference of two int variable on a new line.
   - Print the sum and difference of two float variable rounded to one decimal place on a new line.

Input Format
The first line contains two integers. The second line contains two floating point numbers.
Constraints: $1 \leq$ ***integer variables*** $\leq 10^4$, $1 \leq$ ***float variables*** $\leq 10^4$

Output Format
Print the sum and difference of both integers separated by a space on the first line, and the sum and difference of both float (scaled to ***1*** decimal place) separated by a space on the second line.

Sample Input
10 4
4.0 2.0

Sample Output
14 6
6.0 2.0

Program:

```
#include <stdio.h>
int main()
{
    int a, b;
    float n, m;
    scanf ("%d %d", &a, &b);
    scanf ("%f %f", &n, &m);
    printf ("%d %d", a+b, a-b);
    printf ("\n");
    printf ("%.1f %.1f", n+m, n-m);
    return 0;
}
```

Ex. No.: **Date:**

Average Marks

Problem Statement

Write a program to input a name (as a single character) and marks of three tests as m1, m2, and m3 of a student considering all the three marks have been given in integer format.

Now, you need to calculate the average of the given marks and print it along with the name as mentioned in the output format section.

All the test marks are in integers and hence calculate the average in integer as well. That is, you need to print the integer part of the average only and neglect the decimal part.

Input Format :

Line 1 : Name(Single character)
Line 2 : Marks scored in the 3 tests separated by single space.

Output Format:
First line of output prints the name of the student. Second line of the output prints the average mark.

Constraints
Marks for each student lie in the range 0 to 100 (both inclusive)

Sample Input 1 :
A
3 4 6

Sample Output 1 :
A
4

Program:

```
#include <stdio.h>
int main()
{
    int a, b, c;
    char d;
    scanf("%c\n", &d);
    scanf("%d %d\n", &a, &b, &c);
    printf("%c\n", d);
    printf("%d", (a+b+c)/3);
    return 0;
}
```

Ex. No.: **Date:**

Basic Data Types

Problem Statement:

Some *C* data types, their format specifiers, and their most common bit widths are as follows:

- *Int ("%d"):* 32 Bit integer
- *Long ("%ld"):* 64 bit integer
- *Char ("%c"):* Character type
- *Float ("%f"):* 32 bit real value
- *Double ("%lf"):* 64 bit real value

Reading
To read a data type, use the following syntax: scanf("`format_specifier`", &val)
For example, to read a *character* followed by a *double*: char ch;

```
double d;
scanf("%c %lf", &ch, &d);
```

For the moment, we can ignore the spacing between format specifiers.

Printing
To print a data type, use the following syntax: printf("`format_specifier`", val)
For example, to print a *character* followed by a *double*: char ch = 'd';

```
double d = 234.432;
printf("%c %lf", ch, d);
```

Note: You can also use *cin* and *cout* instead of *scanf* and *printf*; however, if you are taking a million numbers as input and printing a million lines, it is faster to use *scanf* and *printf*.

Input Format
Input consists of the following space-separated values: *int*, *long*, *char*, *float*, and *double*, respectively.

Output Format
Print each element on a new line in the same order it was received as input. Note that the floating-point value should be correct up to 3 decimal places and the double to 9 decimal places.

Sample Input

```
3
12345678912345
a
334.23
14049.30493
```

Sample Output

```
3
12345678912345
a
334.230
14049.304930000
```

Program:

```
#include <stdio.h>
int main()
{
    int a;
    long b;
    char c;
    float d;
    double e;
    scanf("%d %ld %c %f %lf\n",&a,&b,&c,&d,
                                &e);

    printf("%d\n",a);
    printf("%ld\n",b);
    printf("%c\n",c);
    printf("%.3f\n",d);
    printf("%.9lf",e);
    return 0;
}
```

Ex. No.: **Date:**

ASCII Value and Adjacent Characters

Problem Statement:

Write a program to print the ASCII value and the two adjacent characters of the given character.

Input Format: Reads the character

Output Format: First line prints the ascii value, second line prints the previous character and next character of the input character

Sample Input 1:

E

Sample Output 1:

69
D F

Program:

```
#include <stdio.h>
int main()
{
    char c;
    scanf("%c\n",&c);
    printf("%d\n", c);
     printf("%c  %c", (-1), (+1));
     return 0;
}
```

Operators and Expressions, Managing Input and Output Operations

Height Units

Problem Statement:

Many people think about their height in feet and inches, even in some countries that primarily use the metric system. Write a program that reads a number of feet from the user, followed by a number of inches. Once these values are read, your program should compute and display the equivalent number of centimeters.
Hint: One foot is 12 inches. One inch is 2.54 centimeters.

Input Format
First line, read the number of feet.
Second line, read the number of inches.
Output Format
In one line print the height in centimeters.
Note: All of the values should be displayed using two decimal places.

Sample Input 1
5
6

Sample Output 1
167.64

Program:

```
#include <stdio.h>
int main()
{
    int a, b;
    scanf("%d %d", &a, &b);
    float c = ((a*12*2.54)+(b*2.54));
    printf("%.2f", c);
    return 0;
}
```

Ex. No.: **Date:**

Arithmetic

Problem Statement:

Create a program that reads two integers, a and b, from the user. Your program should compute and display:

- The sum of a and b
- The difference when b is subtracted from a
- The product of a and b
- The quotient when a is divided by b
- The remainder when a is divided by b

Input Format
First line, read the first number.
Second line, read the second number.

Output Format
First line, print the sum of a and b
Second line, print the difference when b is subtracted from a
Third line, print the product of a and b
Fourth line, print the quotient when a is divided by b
Fifth line, print the remainder when a is divided by b

Sample Input 1
100
6

Sample Output1
106
94
600
16
4

Program:

```
#include<stdio.h>
int main()
{
    int a, b;
    scanf("%d %d", &a, &b);
    printf("%d\n", (a+b));
    printf("%d\n", (a-b));
    printf("%d\n", (a*b));
    printf("%d\n", (a/b));
    printf("%d", (a%b));
    return 0;
}
```

Ex. No.: **Date:**

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

Ex. No.: **Date:**

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);
    float b = (a*3.49), c=(b* 0.6), d=(b - c);
    printf ("Regular Price : %.2f \n", b);
    printf ("Discount : %.2f \n", c);
    printf ("Total : %.2f", d);
    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 **Yi**(1<=i<=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<=N<=1000000 1<=X,Y<=1000000

SAMPLE INPUT 1
100
110

SAMPLE OUTPUT 1
YES

Program:

```
#include <stdio.h>
int main()
{
    int x, y;
    scanf("%d %d", &x, &y);
    (x<=y)? printf("YES") : printf("NO");
    return 0;
}
```

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
$0 < N < 106$

SAMPLE INPUT 1
1

SAMPLE OUTPUT
0

Program:

```
#include <stdio.h>
int main()
{
    int N;
    scanf("%d", &N);
    printf("%d", (N * (N-1))/2);
    return 0;
}
```

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);
    if ((a > b) && (a>c))
    {
        printf("%d", a);
    }
    else if ((b>a) && (b > c))
    {
        printf("%d",b);
    }
    else
    {
        printf("%d",c);
    }
     return 0;
}
```

Decision Making and Branching – if, if...else, nested if...else, if...else if, Switch-Case

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%10==y%10) {
                printf ("true") ; }
        else {
                printf ("flase");
        }
}
```

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");}
}
```

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;
}
```

Result:

Thus the C-programs are implemented and executed sucessfully.

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:

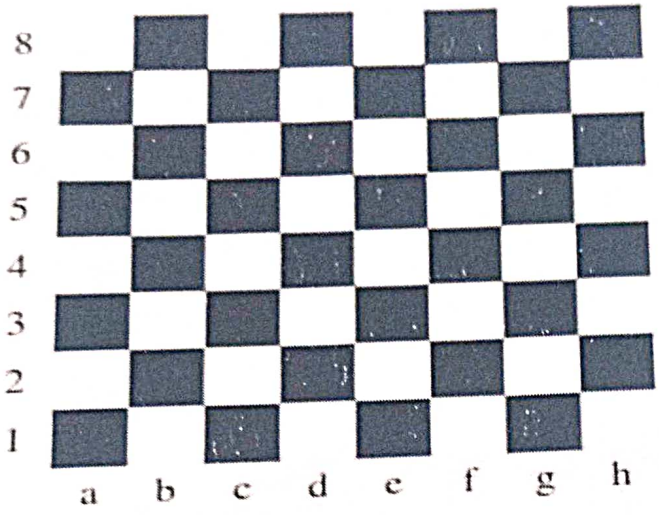
```
#include <stdio.h>
int main()
{
    int year;
    scanf("%d", &year);
    if(year%12==8){ printf("Dragon");}
    else if(year%12==9){ printf("Snake");}
    else if(year%12==10){ printf("Horse");}
    else if(year%12==11){ printf("Sheep");}
    else if(year%12==0){ printf("Monkey");}
    else if(year%12==1){ 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");}
    return 0;
}
```

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.");
    }
}
```

Result:

Hence the C program is implemented and executed successfully.

Date:

GE23131 - Programming Using C

Ex. No. :

Day of Year

Problem Statement:

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

There are two calendars - one for normal years with 365 days, and one for leap
366 days. Leap years are divisible by 4. Centuries, like 1900, are not leap years
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

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", d+31); 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+31+d); break;
        case 10: printf("%d", 31+feb+31+30+31+30+31+31+30+d); break;
        case 11: printf("%d", 31+feb+31+30+31+30+31+31+30+31+d); break;
        case 12: printf("%d", 31+feb+31+30+31+30+31+31+30+31+30+d); break;
    }
}
```

Ex. No. :

Date :

Suppandi & Areas

Problem Statement:

Suppandi is trying to take part in the local village math quiz. In the first round, he is asked about shapes and areas. Suppandi, is confused, he was never any good at math. And also, he is bad at remembering the names of shapes. Instead, you will be helping him calculate the area of shapes.

- When he says rectangle, he is actually referring to a square.
- When he says square, he is actually referring to a triangle.
- When he says triangle, he is referring to a rectangle
- And when he is confused, he just says something random. At this point, all you can do is say 0.

Help Suppandi by printing the correct answer in an integer.

Input Format

- Name of shape (always in upper case R --> Rectangle, S --> Square, T --> Triangle)
- Length of 1 side
- Length of other side

  Note: In case of triangle, you can consider the sides as height and length of base

Output Format

- Print the area of the shape.

Sample Input 1

T
10
20

Sample Output 1

200

Program:

```
#include <stdio.h>
int main()
{
    int a, b;
    char c;
    scanf("%c %d %d", &c, &a, &b);
    switch (c)
    {
        case 'R': printf("%d", a * b); break;
        case 'S': printf("%.0f", (0.5) * a * b); break;
        case 'T': printf("%d", a * b); break;
        default : printf("0"); break;
    }
}
```