PRIYADHARSHINI S
ECE D
240801255

Problem Statement 1:

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

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main(){
    int a,b;
    scanf("%d %d",&a,&b);
    if(a%10==b%10){
        printf("true");

    }
    else{
        printf("false");

    }
    return 0;
}
```

	Input	Expected	Got	
✓	25 53	false	false	✓
✓	27 77	true	true	✓

Passed all tests! ✓

Problem Statement 2:

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

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main(){
    int n;
    scanf("%d",&n);
    if(n%2!=0){
        printf("Weird\n");

    }
    else{
        if(n>=2 && n<=5){
            printf("Not Weird\n");
        }
        else if(n>=6 && n<=20){
            printf("Weird\n");
        }
        else if(n>20){
            printf("Not Weird\n");
        }
    }
    return 0;
}
```

	Input	Expected	Got	
✓	3	Weird	Weird	✓
✓	24	Not Weird	Not Weird	✓

Passed all tests! ✓

Problem Statement 3:

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

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main(){
    int a,b,c;
    scanf("%d %d %d",&a,&b,&c);
    int largest =a>b?(a>c?a:c):(b>c?b:c);
    int sum_square=a*a+b*b+c*c-largest*largest;
    if(sum_square==largest*largest){
        printf("yes\n");
    }
    else{
        printf("no\n");
    }
    return 0;
}
```

	Input	Expected	Got	
✓	3 5 4	yes	yes	✓
✓	5 8 2	no	no	✓

Passed all tests! ✓

Sample

Output 1

Yes

Problem Statement 4:

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.

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main(){
    int side;
    scanf("%d",&side);
    if(side>=3 && side<=10){
        switch (side){
            case 3:
             printf("Triangle\n");
             break;

            case 4:
            printf("Quadrilateral\n");
            break;
            case 5:
            printf("Pentagon\n");
            break;
            case 6:
            printf("Hexagon\n");
            break;
            case 7:
            printf("Heptagon\n");
            break;
            case 8:
            printf("Octogon");
            break;
            case 9:
            printf("Nonagon");
            break;
            case 10:
            printf("Decagon");
            break;
        }
    }
```

```
32          }
33      }
34      else{
35              printf("The number of sides is not supported.\n");
36      }
37      return 0;
38  }
```

	Input	Expected	Got	
✓	3	Triangle	Triangle	✓
✓	7	Heptagon	Heptagon	✓
✓	11	The number of sides is not supported.	The number of sides is not supported.	✓

Passed all tests! ✓

Problem Statement 5:

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

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main(){
    int year;
    char*animals[]={"Dragon","Snake","Horse","Sheep","Monkey","Roaster","Dog","Pig","Rat","Ox","Tiger","Hare"
    scanf("%d",&year);
    int index=(year-2000)%12;
    printf("%s\n",animals[index]);
    return 0;

}
```

	Input	Expected	Got	
✓	2004	Monkey	Monkey	✓
✓	2010	Tiger	Tiger	✓

Passed all tests! ✓

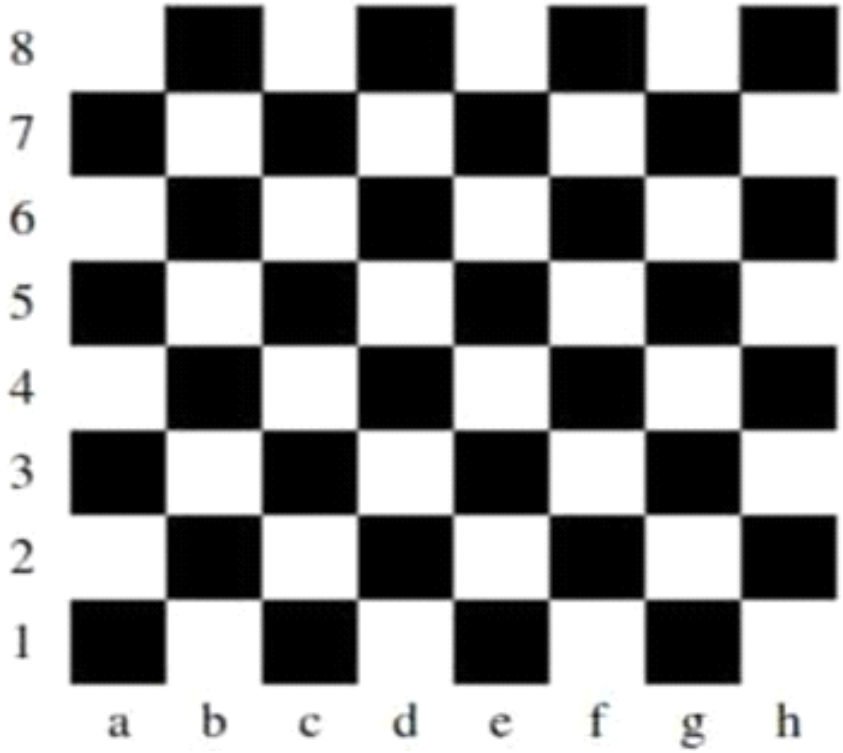
Sample

Output 2

Tiger

Problem Statement 6:

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

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
#include<ctype.h>
int main(){
    char column,row;
    scanf("%c %c",&column,&row);
    column=tolower(column);
    int intial_color=(column=='a'||column=='h')?1:0;
    int square_color=(intial_color+(row-'1'))%2;
    if(square_color==1){
        printf("The square is black.\n");
    }
    else{
        printf("The square is white.\n");
    }
    return 0;
}
```

	Input	Expected	Got	
✓	a 1	The square is black.	The square is black.	✓
✓	d 5	The square is white.	The square is white.	✓

Passed all tests! ✓

Sample

Output 2 The

square is

white.

Problem Statement 7:

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

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main(){
    int date,month,year;
    int days[]={31,28,31,30,31,30,31,31,30,31,30,31};
    scanf("%d %d %d",&date,&month,&year);
    if((year%4==0&&year%100!=0)||(year%400==0)){
        days[1]=29;
    }
    int DOT=date;
    for(int i=0;i<(month-1);i++){
        DOT += days[i];
    }
    printf("%d\n",DOT);
    return 0;
}
```

	Input	Expected	Got	
✓	18 6 2020	170	170	✓

Passed all tests! ✓

Problem Statement 8:

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

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main(){
    char shape;
    int side1,side2;
    scanf("%c %d %d",&shape,&side1,&side2);
    int area;
    switch(shape){
        case'R':
        area=side1*side2;
        break;
        case 'S':
        area=(side1*side2)/2;
        break;
        case 'T':
        area=side1*side2;
        break;
        default:
        area=0;
    }
    printf("%d\n",area);
    return 0;
}
```

	Input	Expected	Got	
✓	T 10 20	200	200	✓
✓	S 30 40	600	600	✓
✓	B 2 11	0	0	✓
✓	R 10 30	300	300	✓
✓	S 40 50	1000	1000	✓

Passed all tests! ✓

Problem Statement 9:

Superman is planning a journey to his home planet. It is very important for him to know which day he arrives there. They don't follow the 7-day week like us. Instead, they follow a 10-day week with the following days:

Day Number Name of Day

- **Sunday**
- **Monday**
- **Tuesday**
- **Wednesday**

- **Thursday**
- **Friday**
- **Saturday**
- **Kryptonday**
- **Coluday**
- **Daxamday**

Here are the rules of the calendar:

- **The calendar starts with Sunday always.**
- **It has only 296 days. After the 296th day, it goes back to Sunday.**

You begin your journey on a Sunday and will reach after n. You have to tell on which day you will arrive when you reach there.

Input format:

- **Contain a number**

n (0 < n) Output

format:

Print the name of the day you are arriving on

Sampl

e Input

7

Sample

Output

Krypton

day

Sampl

e Input

1

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main(){
    int n;
    const char *days[]={"Sunday","Monday","Tuesday","Wednesday","Thursday","Friday","Saturday","Kryptonday","Colu
    scanf("%d",&n);
    int arrival_day=(n%296)%10;
    printf("%s\n",days[arrival_day]);
    return 0;
}
```

	Input	Expected	Got	
✓	7	Kryptonday	Kryptonday	✓
✓	1	Monday	Monday	✓

Passed all tests! ✓

Sample

Output

Monday