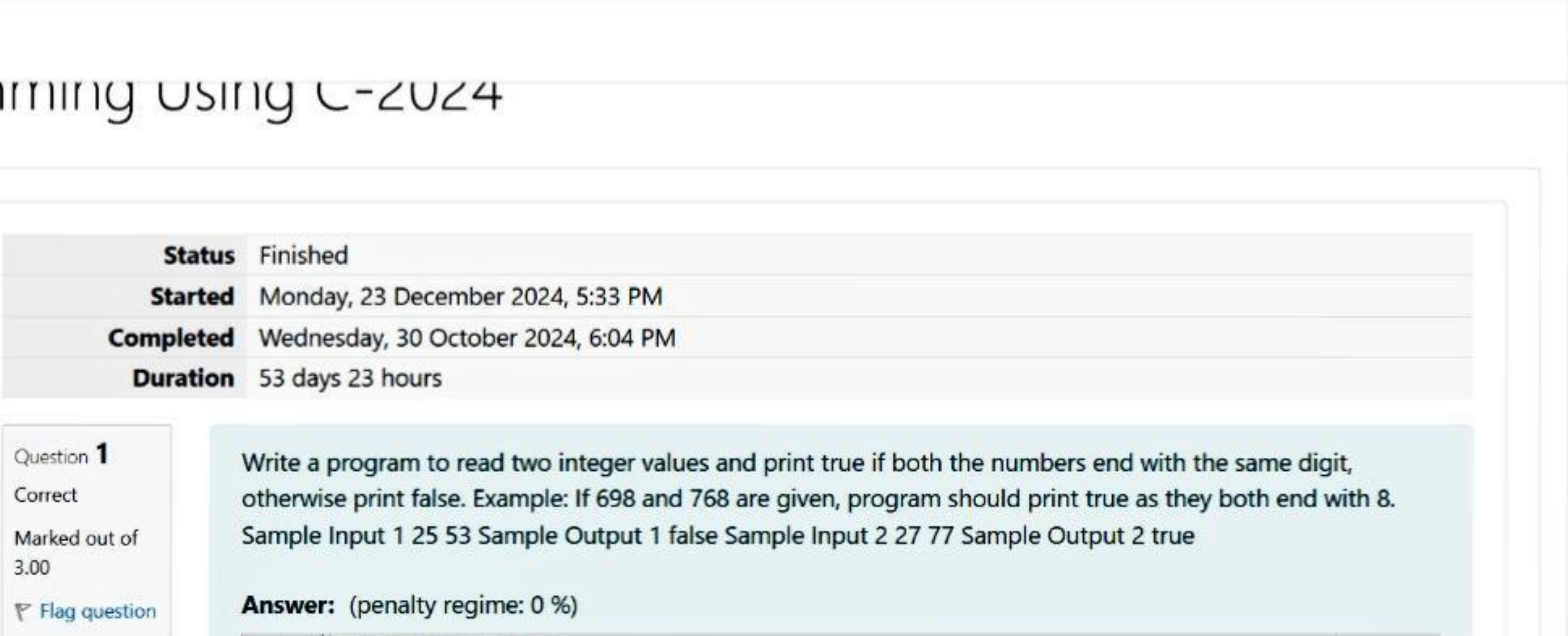


mming Using C-2024

Status	Finished
Started	Monday, 23 December 2024, 5:33 PM
Completed	Wednesday, 30 October 2024, 6:04 PM
Duration	53 days 23 hours

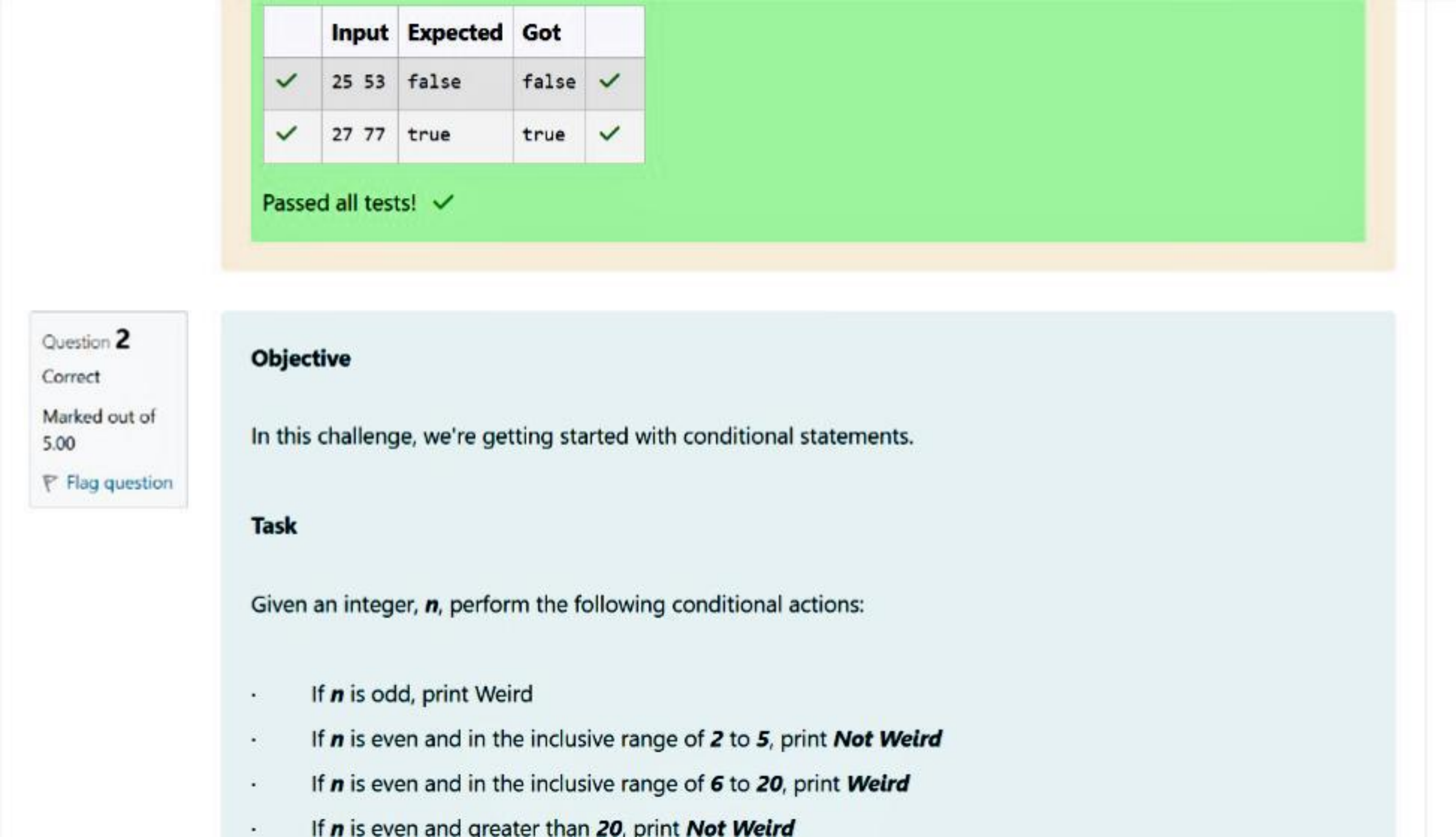
Question **1**

Correct

Marked out of 3.00

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 %)



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

Passed all tests! ✓

Question **2**

Correct

Marked out of 5.00

Flag question

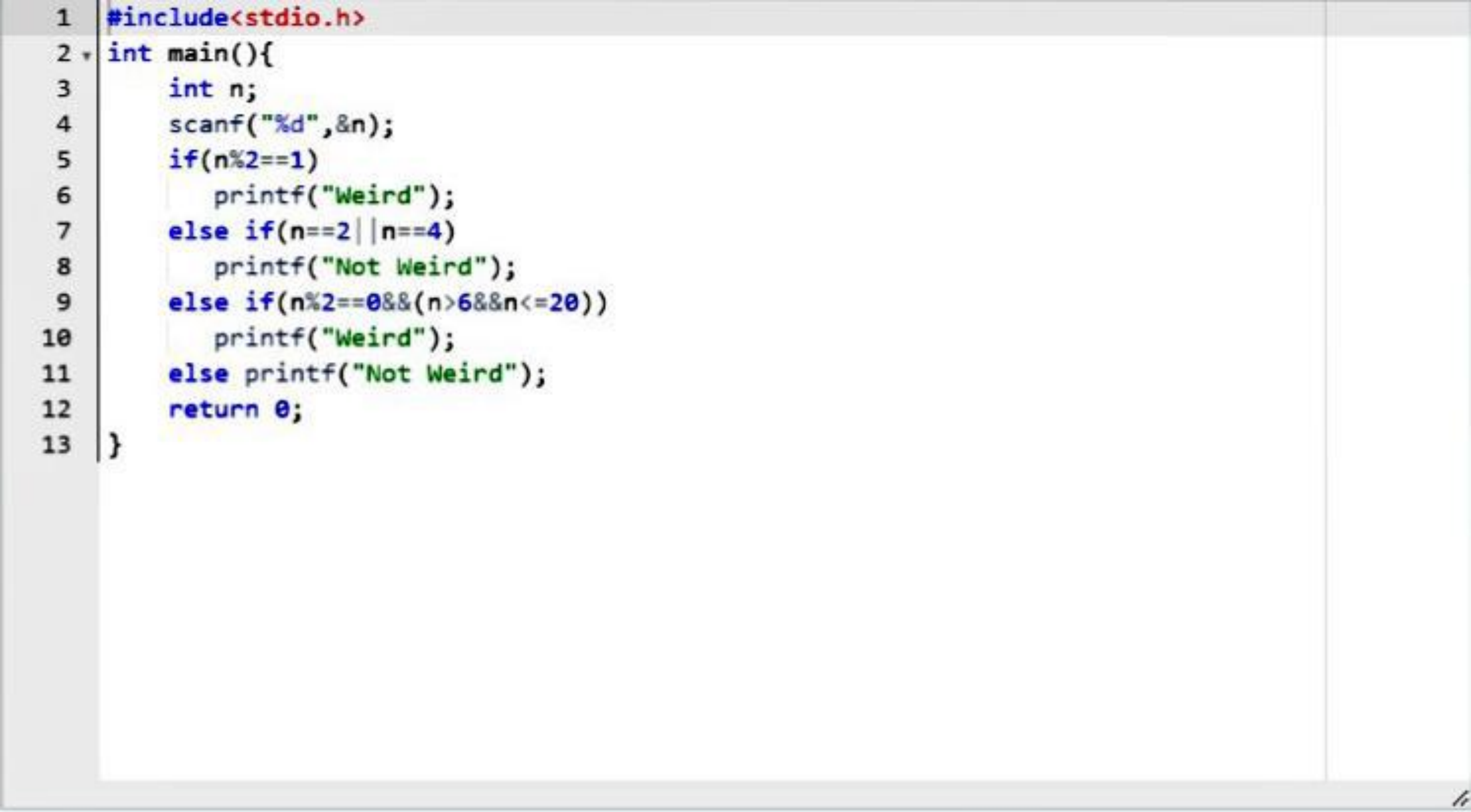
Objective

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

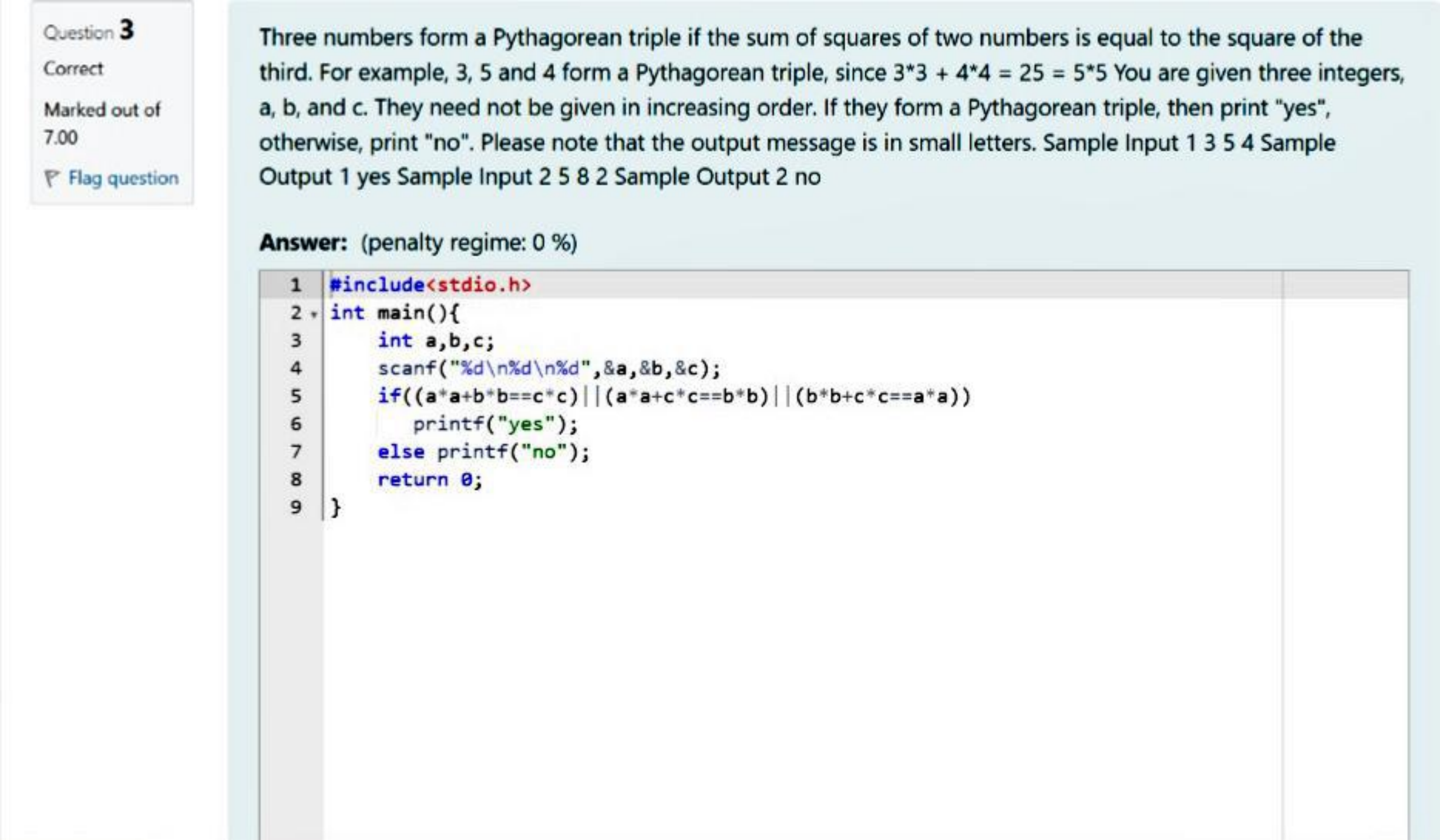
Task

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***



```c
#include<stdio.h>
int main(){
    int n;
    scanf("%d",&n);
    if(n%2==1)
       printf("Weird");
    else if(n==2||n==4)
       printf("Not Weird");
    else if(n%2==0&&(n>6&&n<=20))
       printf("Weird");
    else printf("Not Weird");
    return 0;
}
```



Question 3

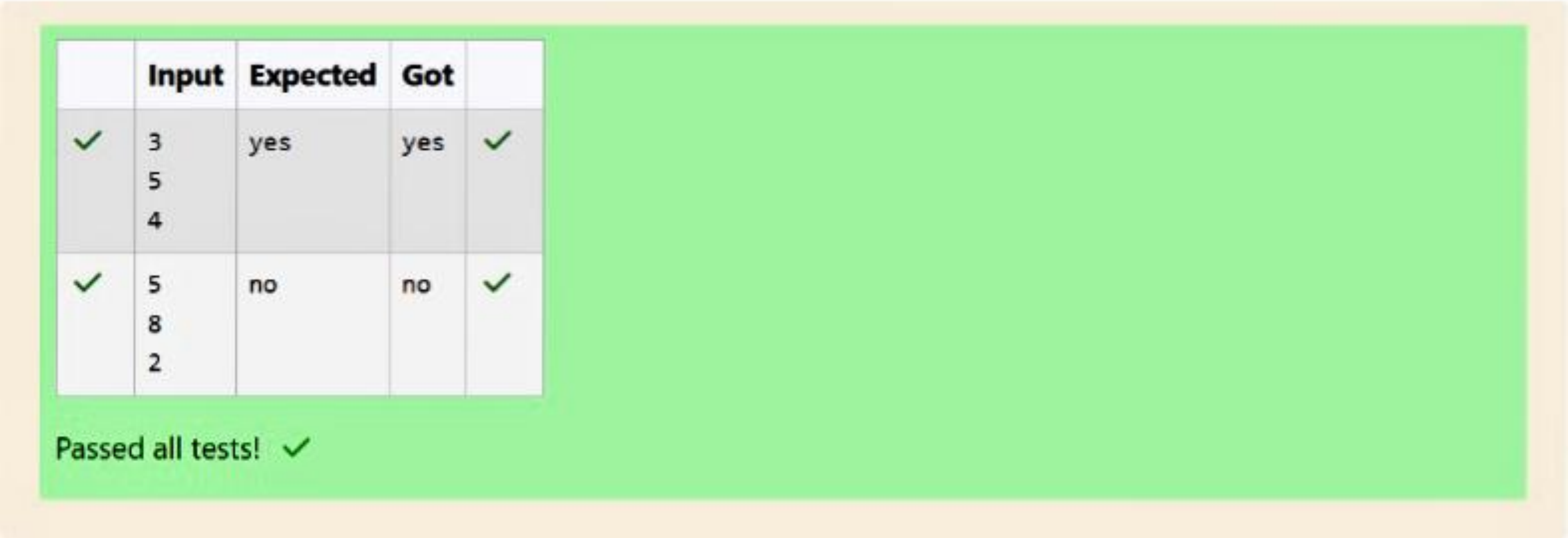
Correct

Marked out of 7.00

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 Sample Input 2 5 8 2 Sample Output 2 no

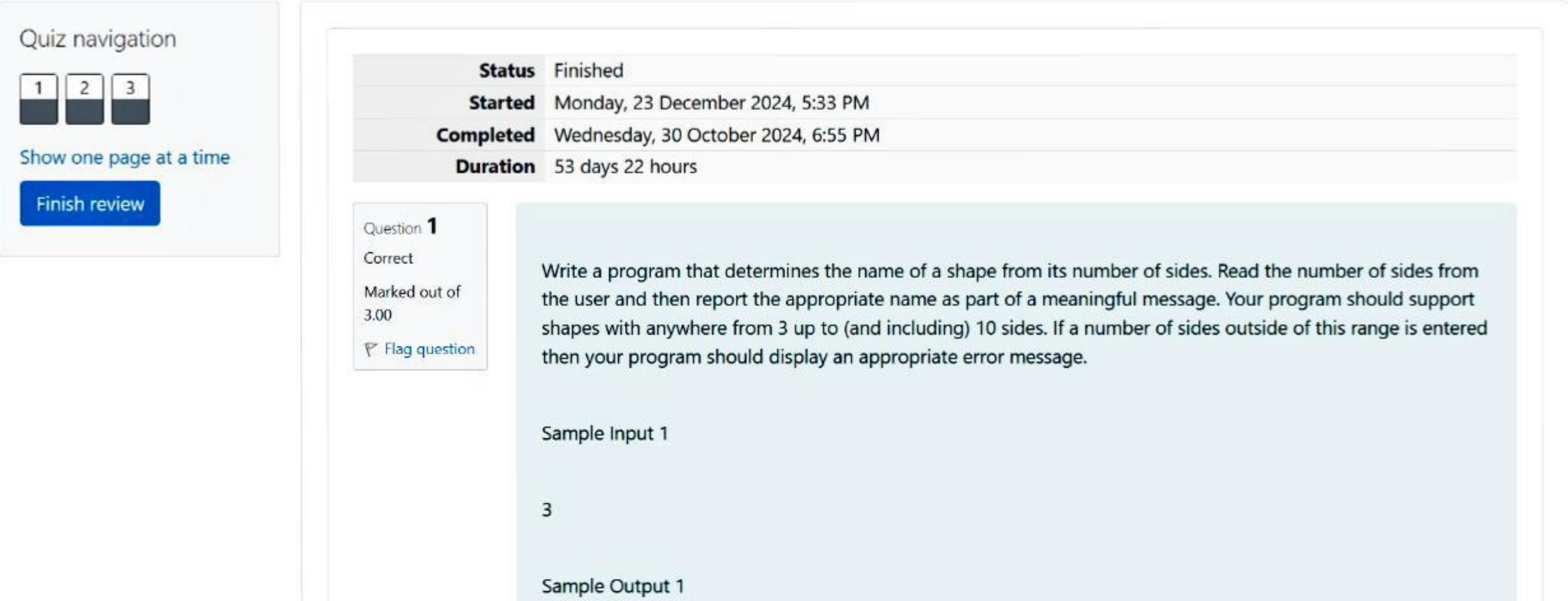
Answer: (penalty regime: 0 %)

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



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

Passed all tests! ✓



Status	Finished
Started	Monday, 23 December 2024, 5:33 PM
Completed	Wednesday, 30 October 2024, 6:55 PM
Duration	53 days 22 hours

Question **1**

Correct

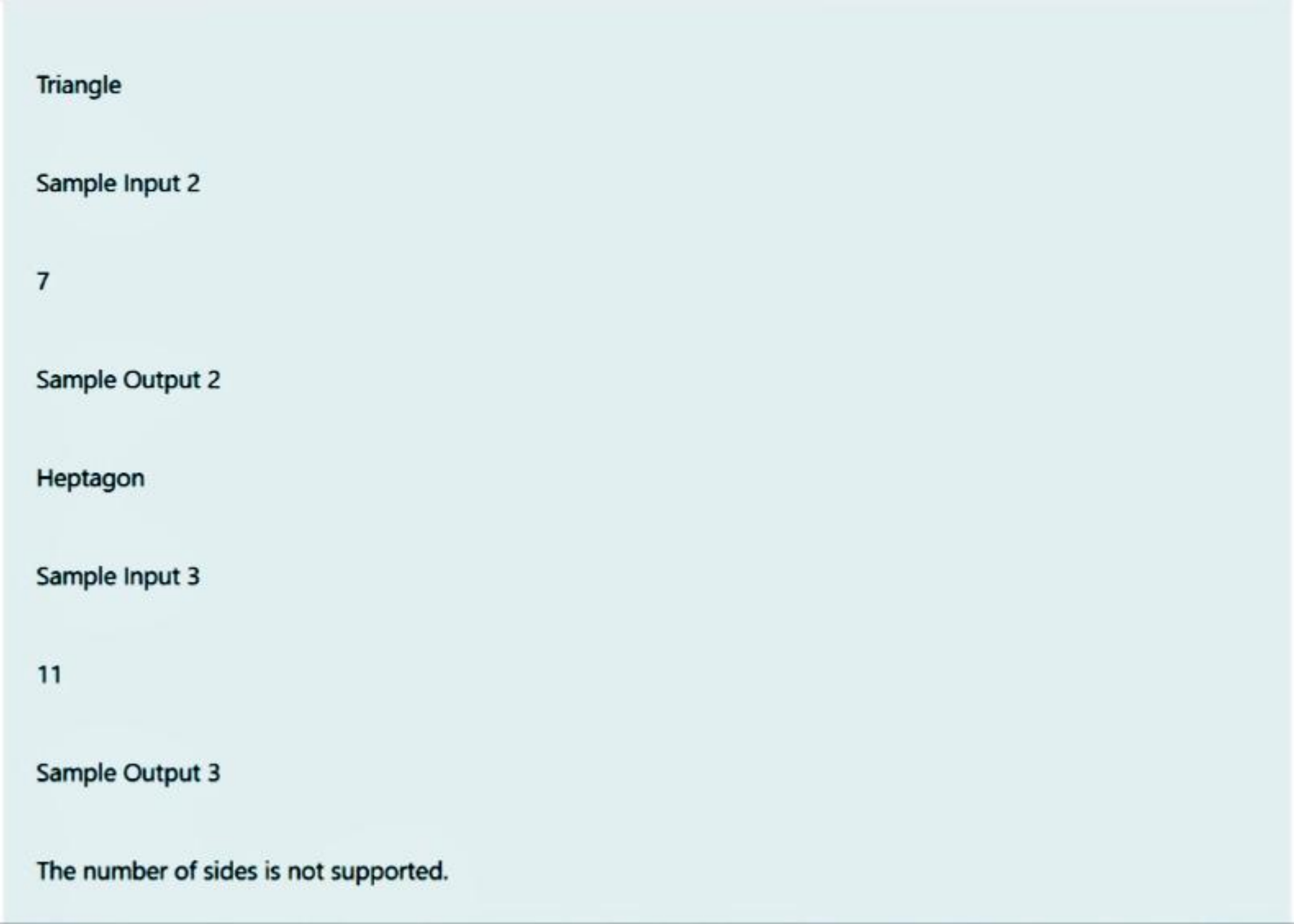
Marked out of 3.00

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
```

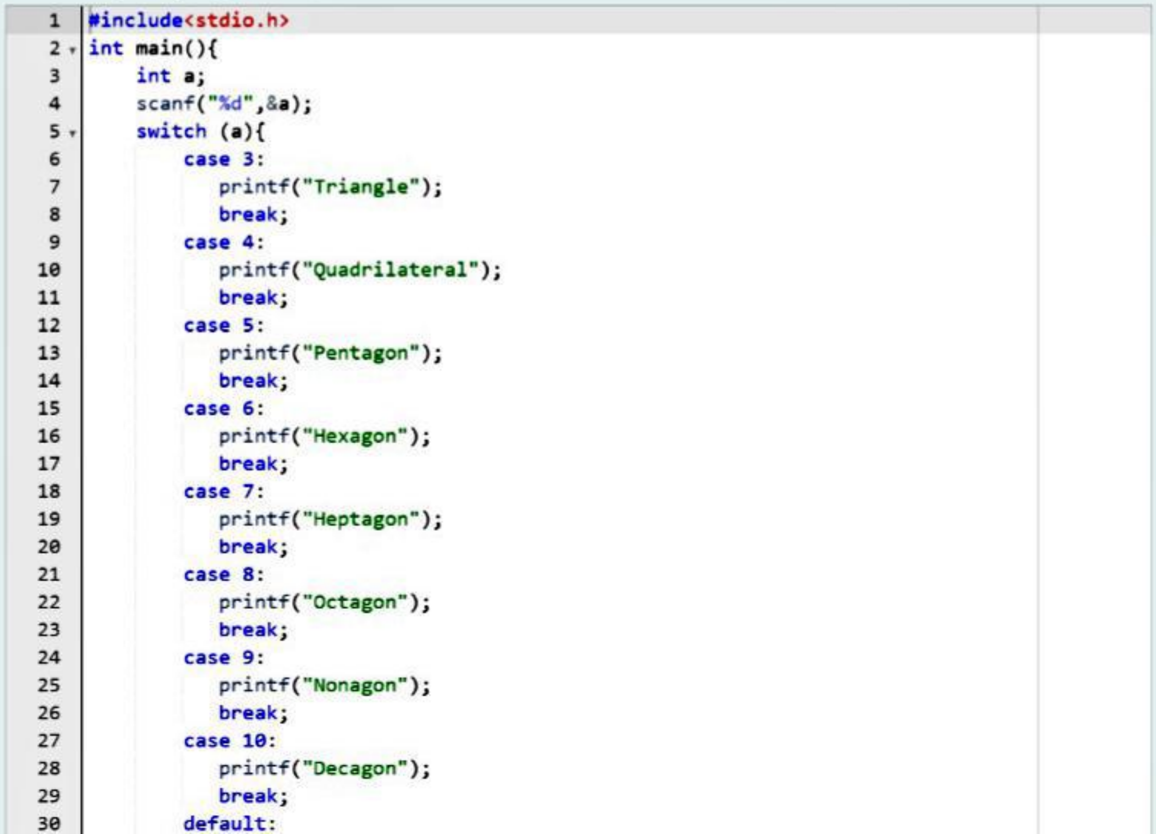
Sample Input 3

```
11
```

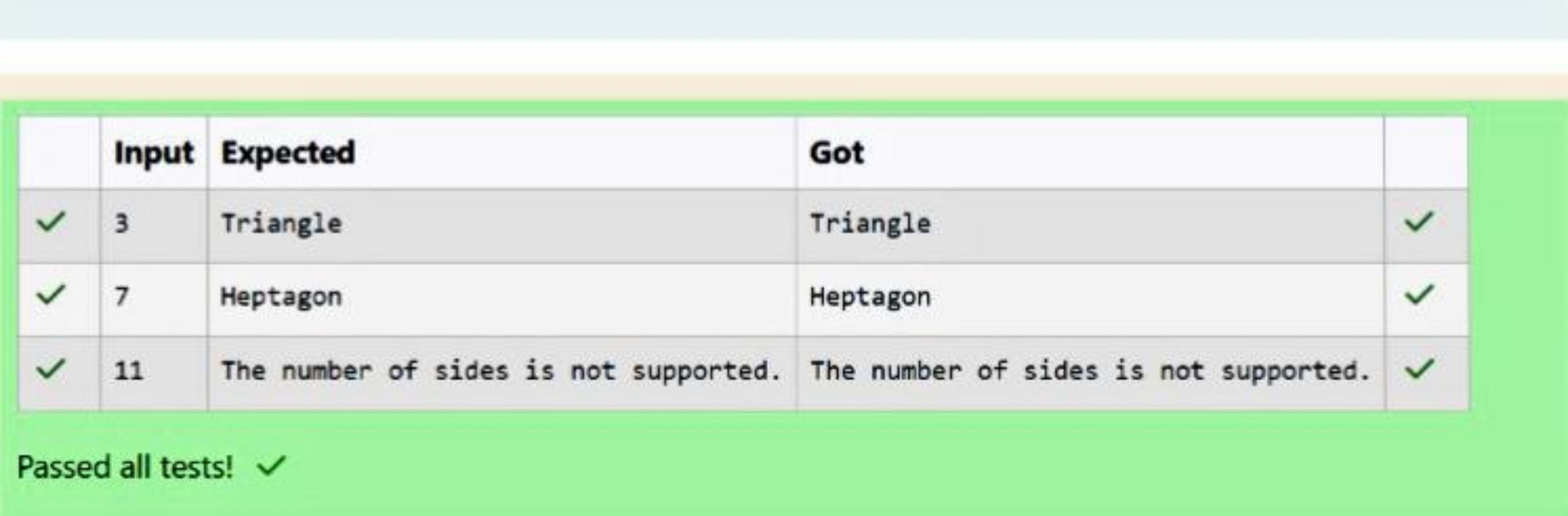
Sample Output 3

```
The number of sides is not supported.
```

Answer: (penalty regime: 0 %)

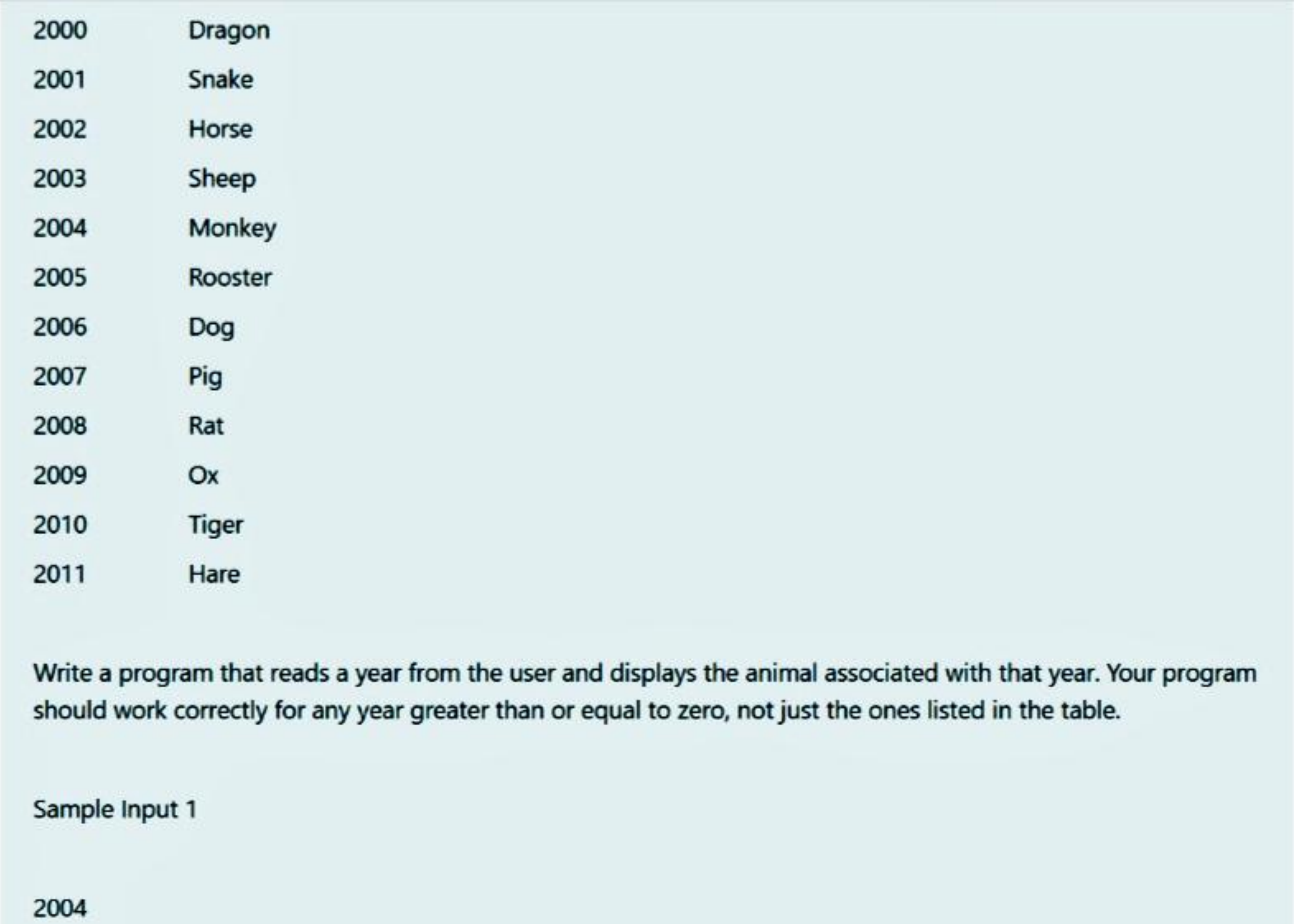


```
#include<stdio.h>
int main(){
    int a;
    scanf("%d",&a);
    switch (a){
        case 3:
           printf("Triangle");
           break;
        case 4:
           printf("Quadrilateral");
           break;
        case 5:
           printf("Pentagon");
           break;
        case 6:
           printf("Hexagon");
           break;
        case 7:
           printf("Heptagon");
           break;
        case 8:
           printf("Octagon");
           break;
        case 9:
           printf("Nonagon");
           break;
        case 10:
           printf("Decagon");
           break;
        default:
```



	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! ✓

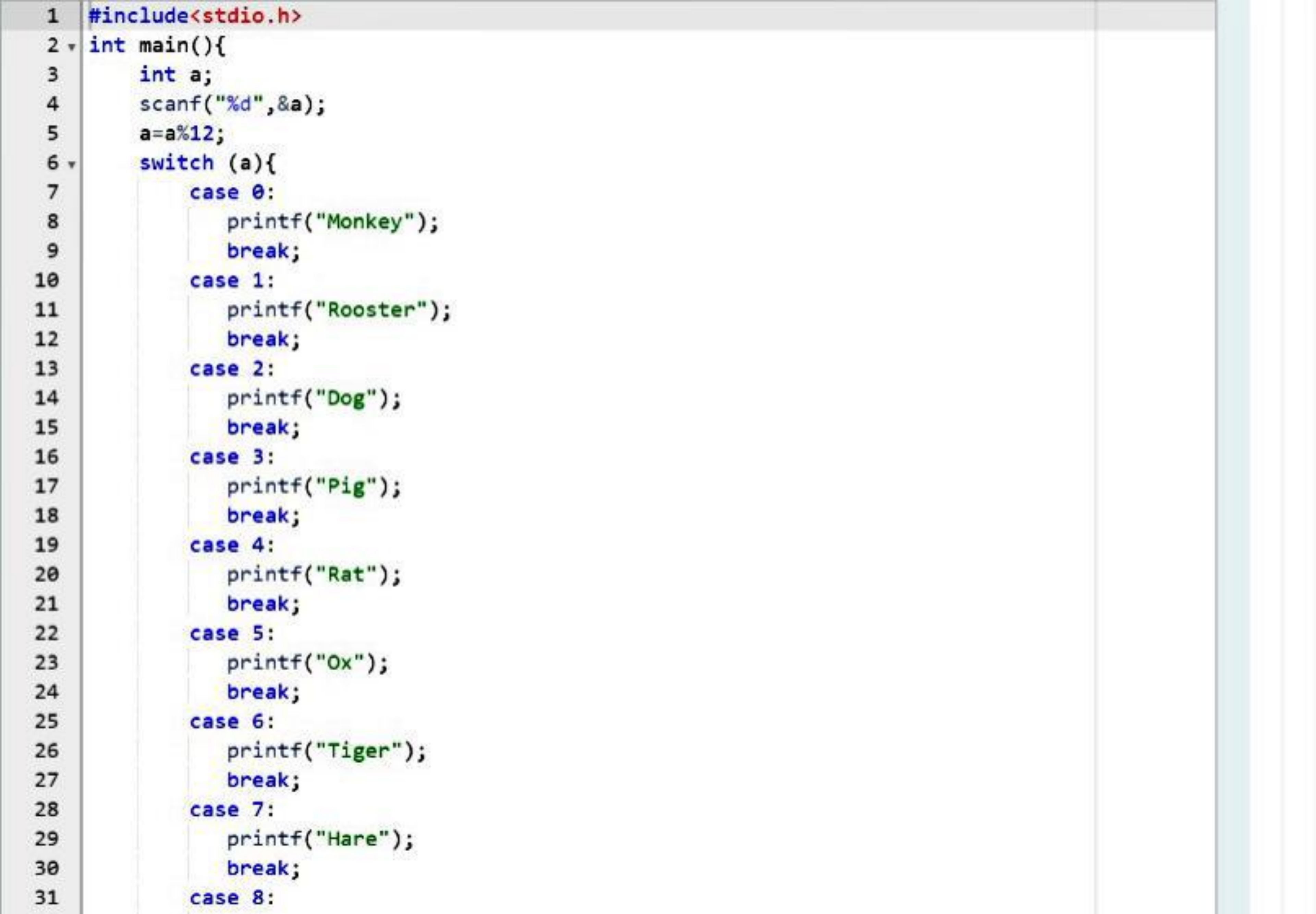


2000	Dragon
2001	Snake
2002	Horse
2003	Sheep
2004	Monkey
2005	Rooster
2006	Dog
2007	Pig
2008	Rat
2009	Ox
2010	Tiger
2011	Hare

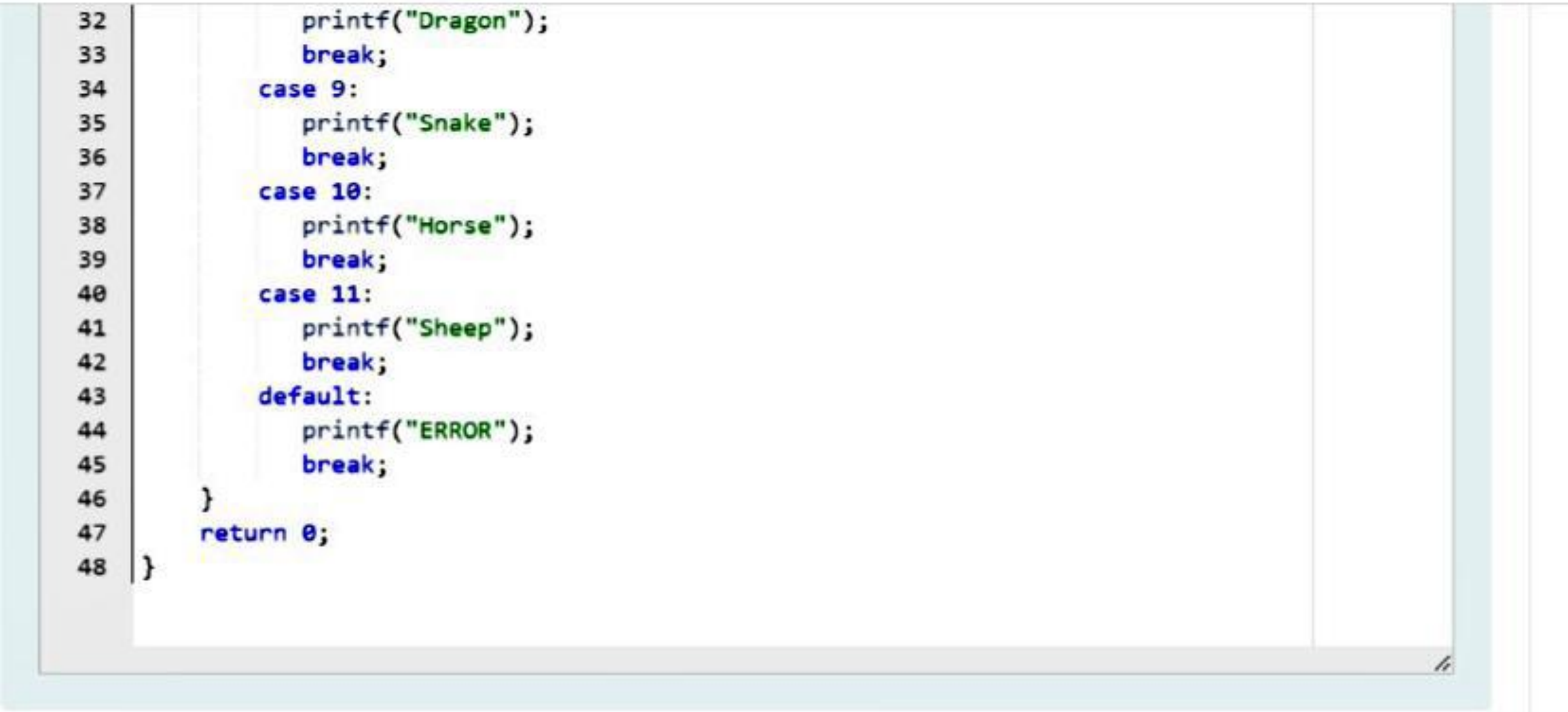
Write a program that reads a year from the user and displays the animal associated with that year. Your program should work correctly for any year greater than or equal to zero, not just the ones listed in the table.

Sample Input 1

```
2004
```



```c
#include<stdio.h>
int main(){
    int a;
    scanf("%d",&a);
    a=a%12;
    switch (a){
        case 0:
            printf("Monkey");
            break;
        case 1:
            printf("Rooster");
            break;
        case 2:
            printf("Dog");
            break;
        case 3:
            printf("Pig");
            break;
        case 4:
            printf("Rat");
            break;
        case 5:
            printf("Ox");
            break;
        case 6:
            printf("Tiger");
            break;
        case 7:
            printf("Hare");
            break;
        case 8:
```



```c
           printf("Dragon");
           break;
        case 9:
           printf("Snake");
           break;
        case 10:
           printf("Horse");
           break;
        case 11:
           printf("Sheep");
           break;
        default:
           printf("ERROR");
           break;
    }
    return 0;
}
```

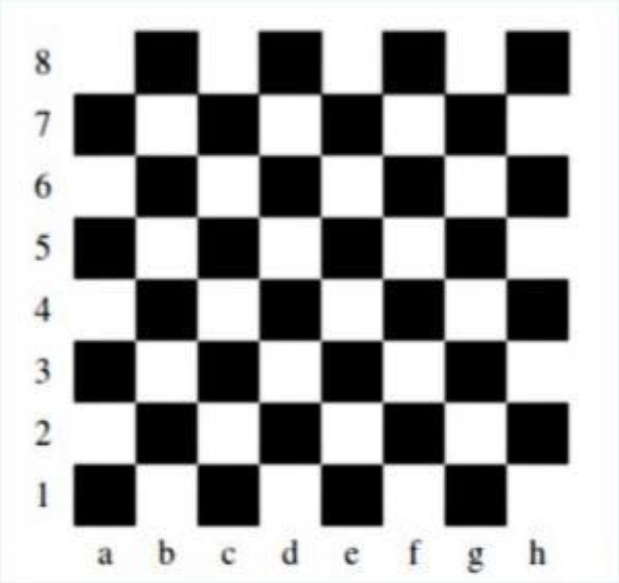
Question **3**

Correct

Marked out of 7.00

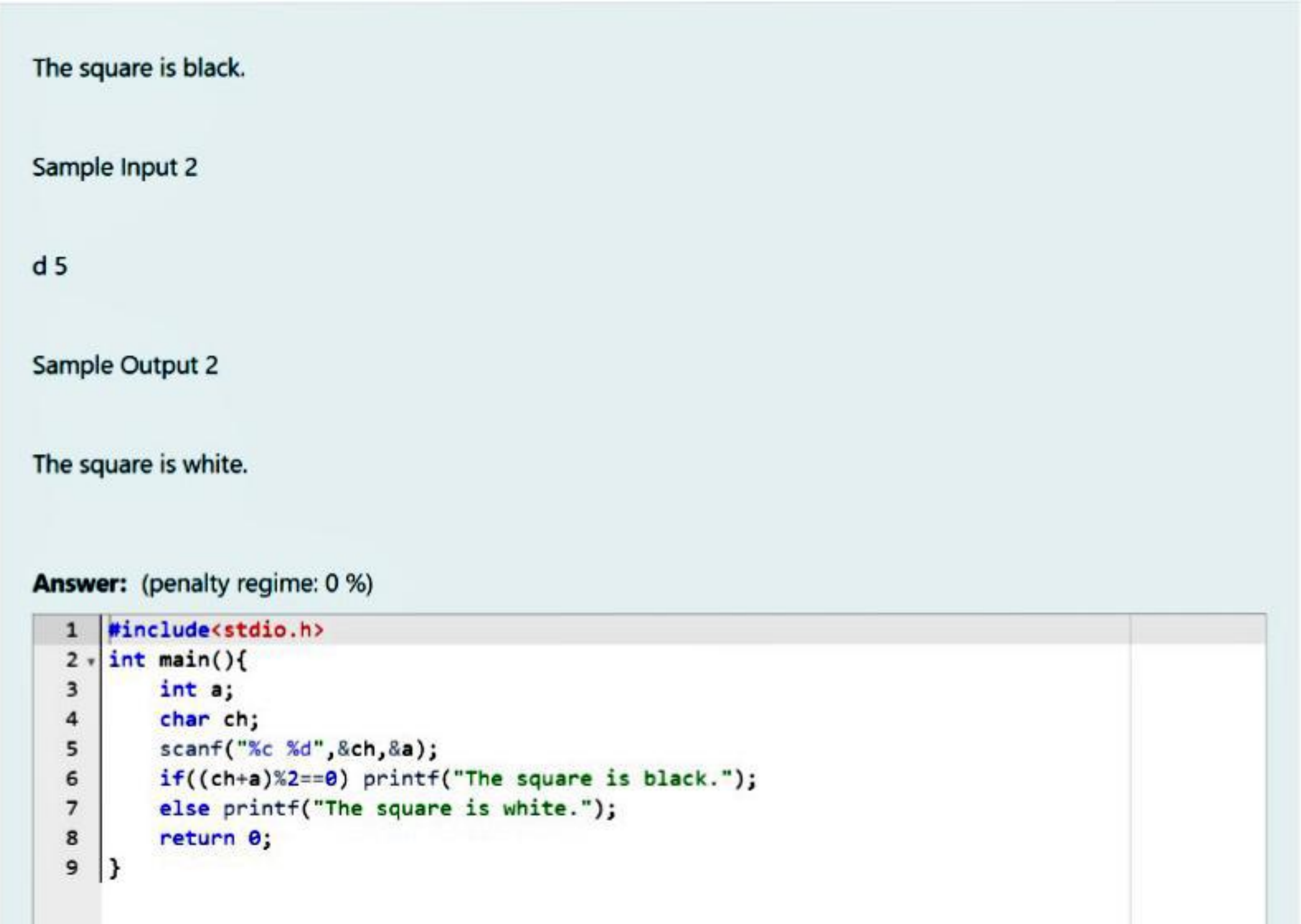
Flag question

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



The square is black.

Sample Input 2

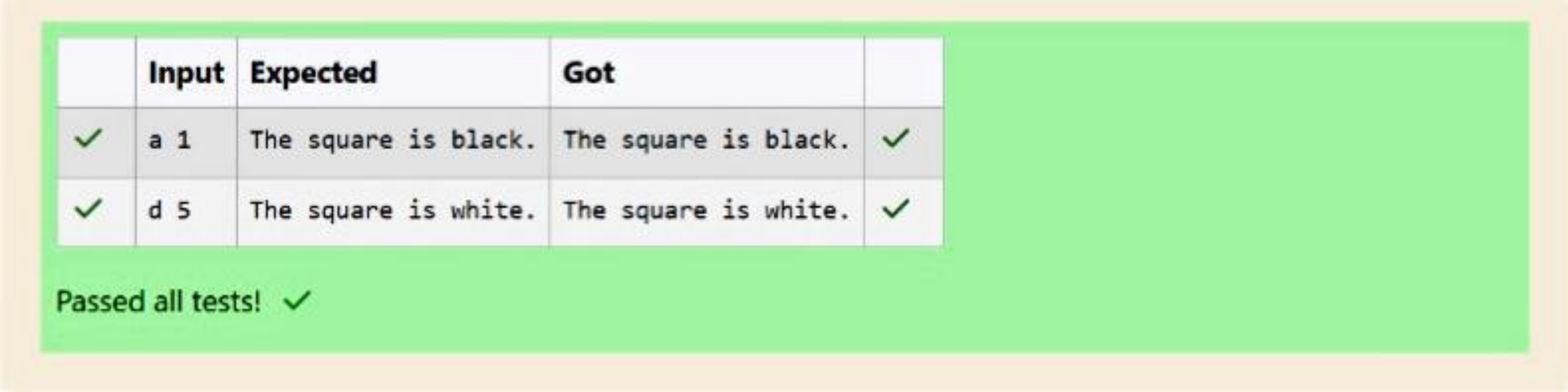
d 5

Sample Output 2

The square is white.

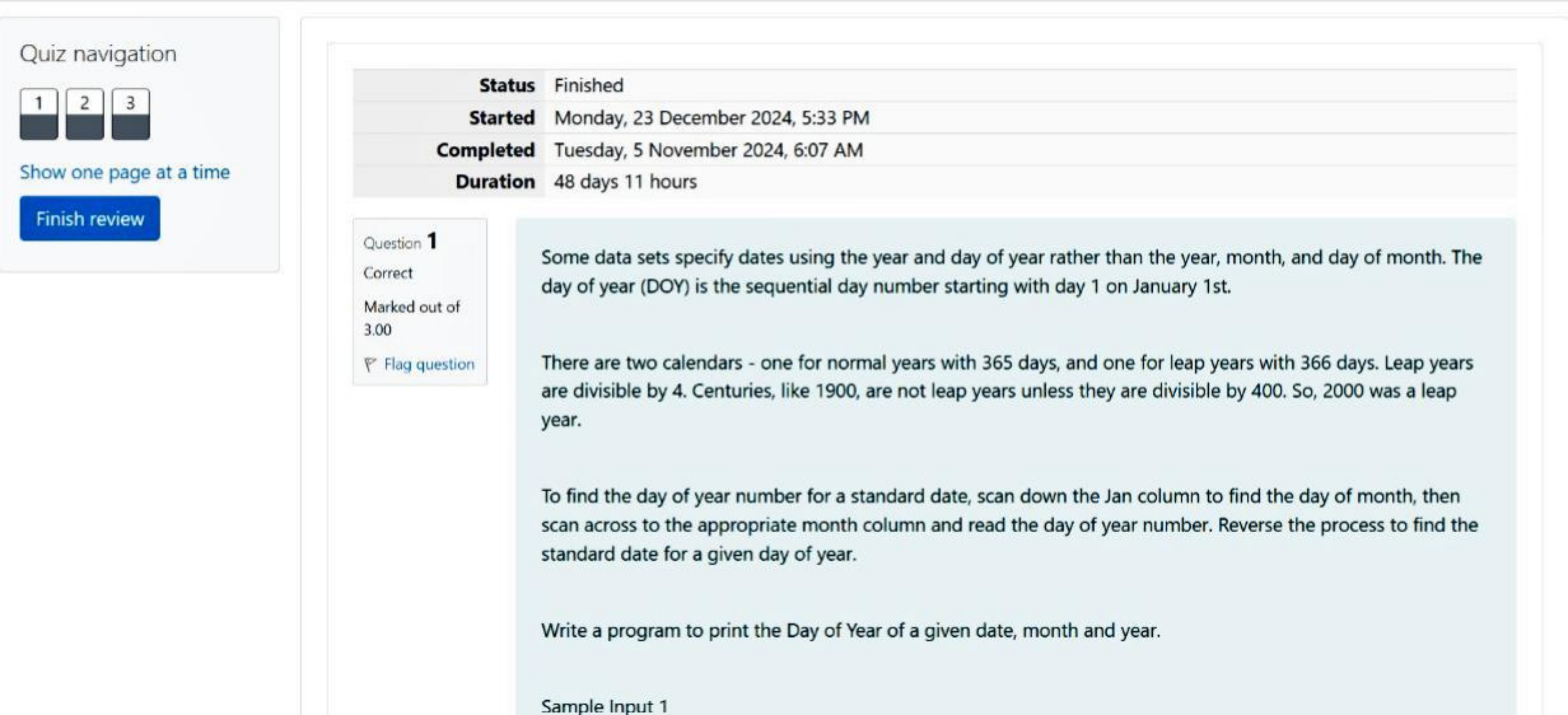
Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main(){
    int a;
    char ch;
    scanf("%c %d",&ch,&a);
    if((ch+a)%2==0) printf("The square is black.");
    else printf("The square is white.");
    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! ✓



Status	Finished
Started	Monday, 23 December 2024, 5:33 PM
Completed	Tuesday, 5 November 2024, 6:07 AM
Duration	48 days 11 hours

Question **1**

Correct

Marked out of 3.00

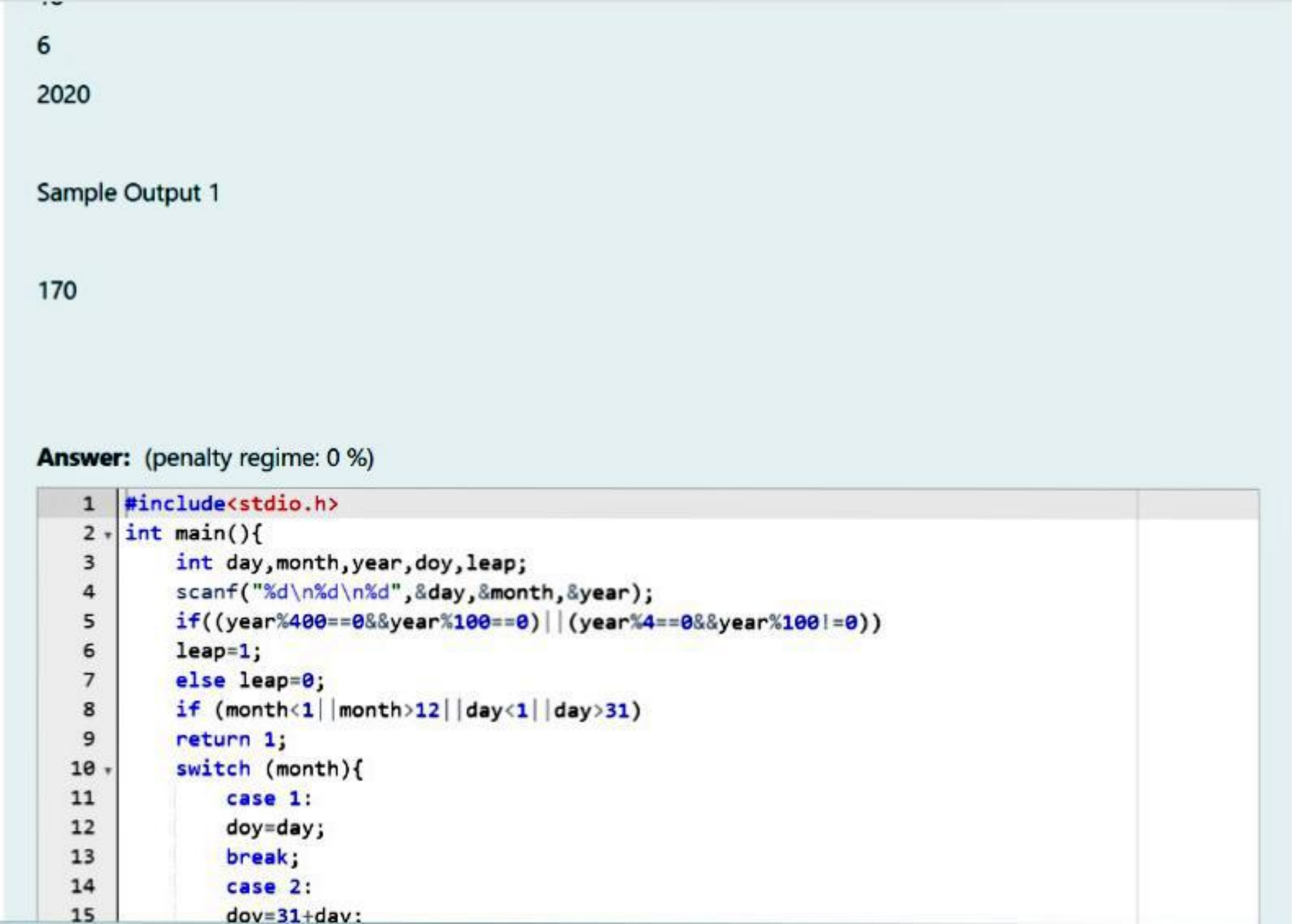
Some data sets specify dates using the year and day of year rather than the year, month, and day of month. The day of year (DOY) is the sequential day number starting with day 1 on January 1st.

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



6

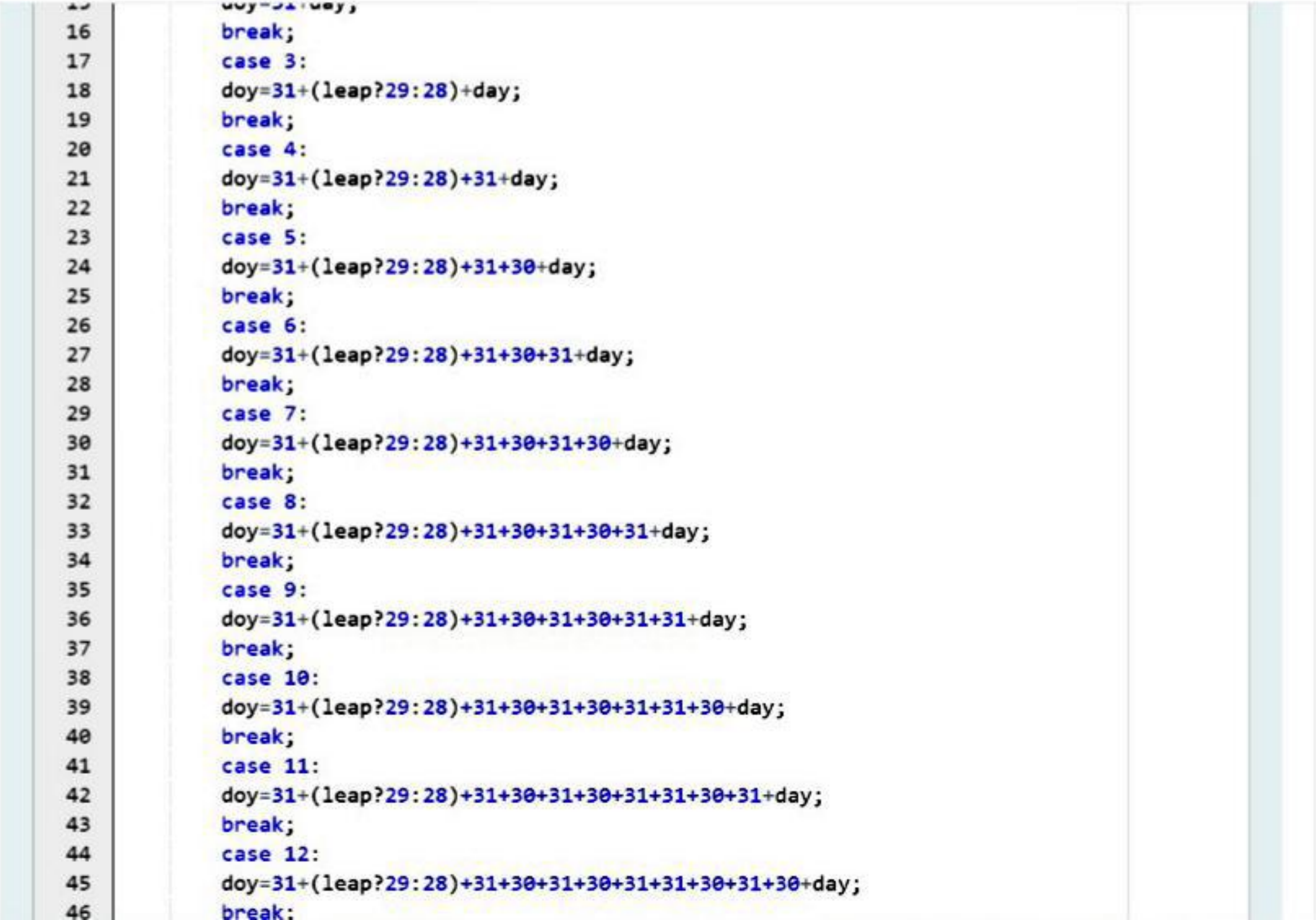
2020

Sample Output 1

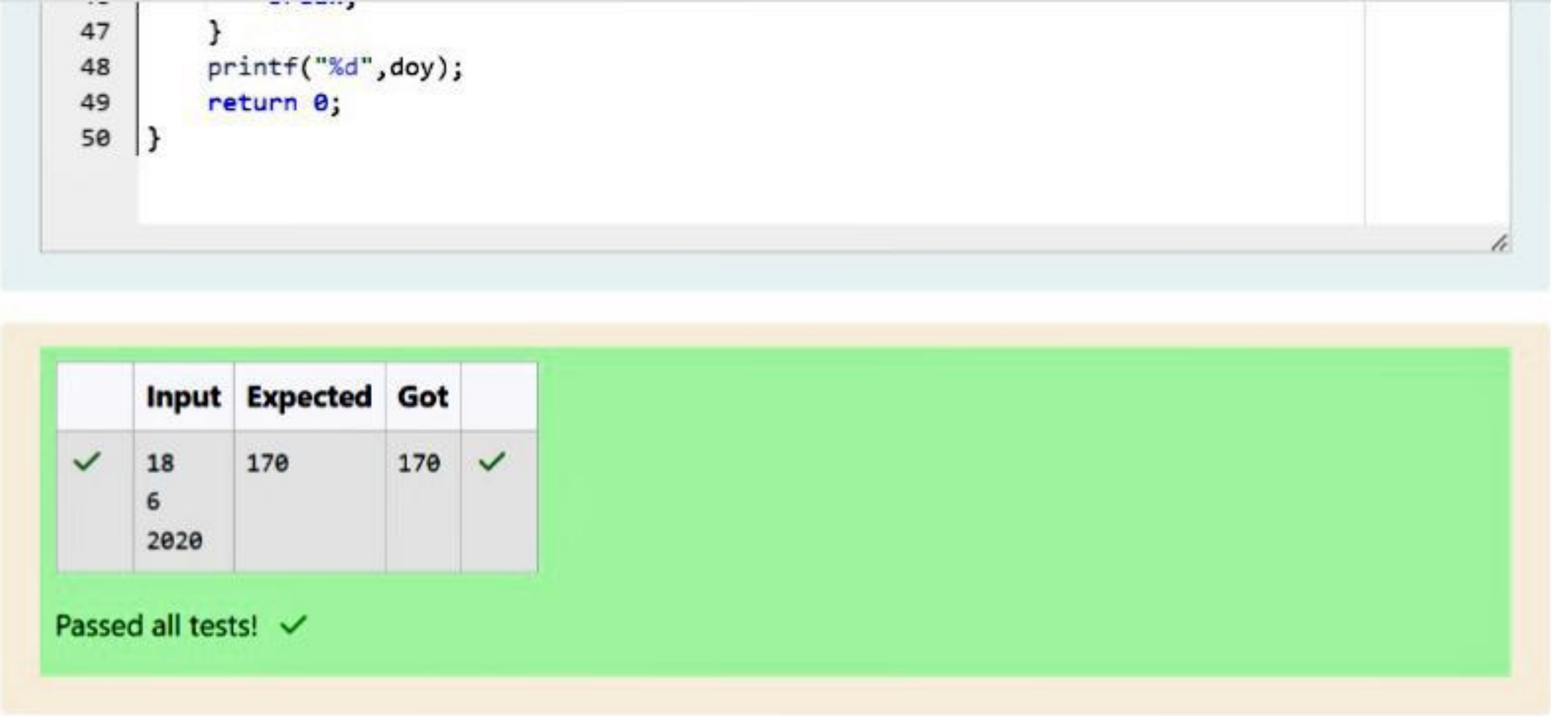
170

Answer: (penalty regime: 0 %)

```
#include<stdio.h>
int main(){
    int day,month,year,doy,leap;
    scanf("%d\n%d\n%d",&day,&month,&year);
    if((year%400==0&&year%100==0)||(year%4==0&&year%100!=0))
    leap=1;
    else leap=0;
    if (month<1||month>12||day<1||day>31)
    return 1;
    switch (month){
        case 1:
        doy=day;
        break;
        case 2:
        doy=31+day;
```



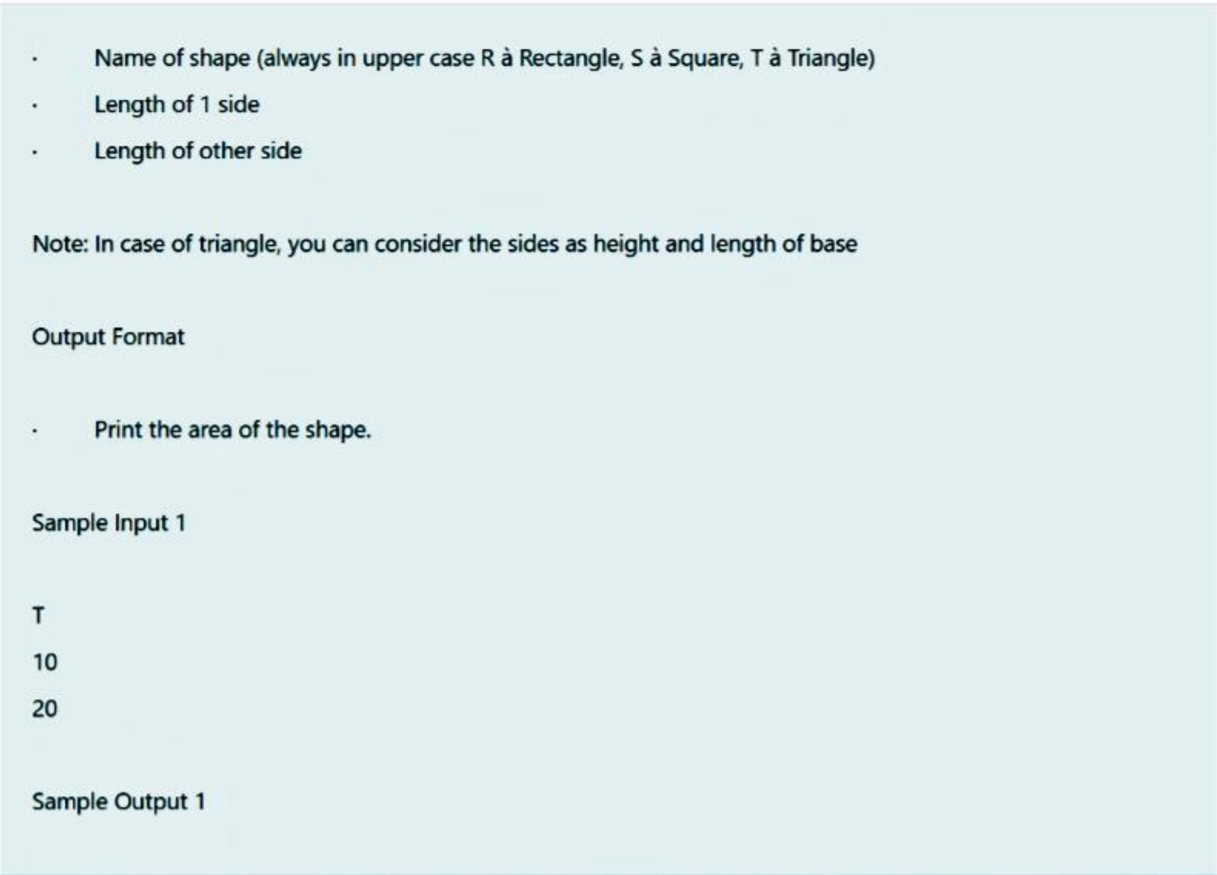
```
doy=31+day;
break;
case 3:
doy=31+(leap?29:28)+day;
break;
case 4:
doy=31+(leap?29:28)+31+day;
break;
case 5:
doy=31+(leap?29:28)+31+30+day;
break;
case 6:
doy=31+(leap?29:28)+31+30+31+day;
break;
case 7:
doy=31+(leap?29:28)+31+30+31+30+day;
break;
case 8:
doy=31+(leap?29:28)+31+30+31+30+31+day;
break;
case 9:
doy=31+(leap?29:28)+31+30+31+30+31+31+day;
break;
case 10:
doy=31+(leap?29:28)+31+30+31+30+31+31+30+day;
break;
case 11:
doy=31+(leap?29:28)+31+30+31+30+31+31+30+31+day;
break;
case 12:
doy=31+(leap?29:28)+31+30+31+30+31+31+30+31+30+day;
break;
```



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

	Input	Expected	Got	
✓	18 6 2020	170	170	✓

Passed all tests! ✓



- Name of shape (always in upper case R à Rectangle, S à Square, T à 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

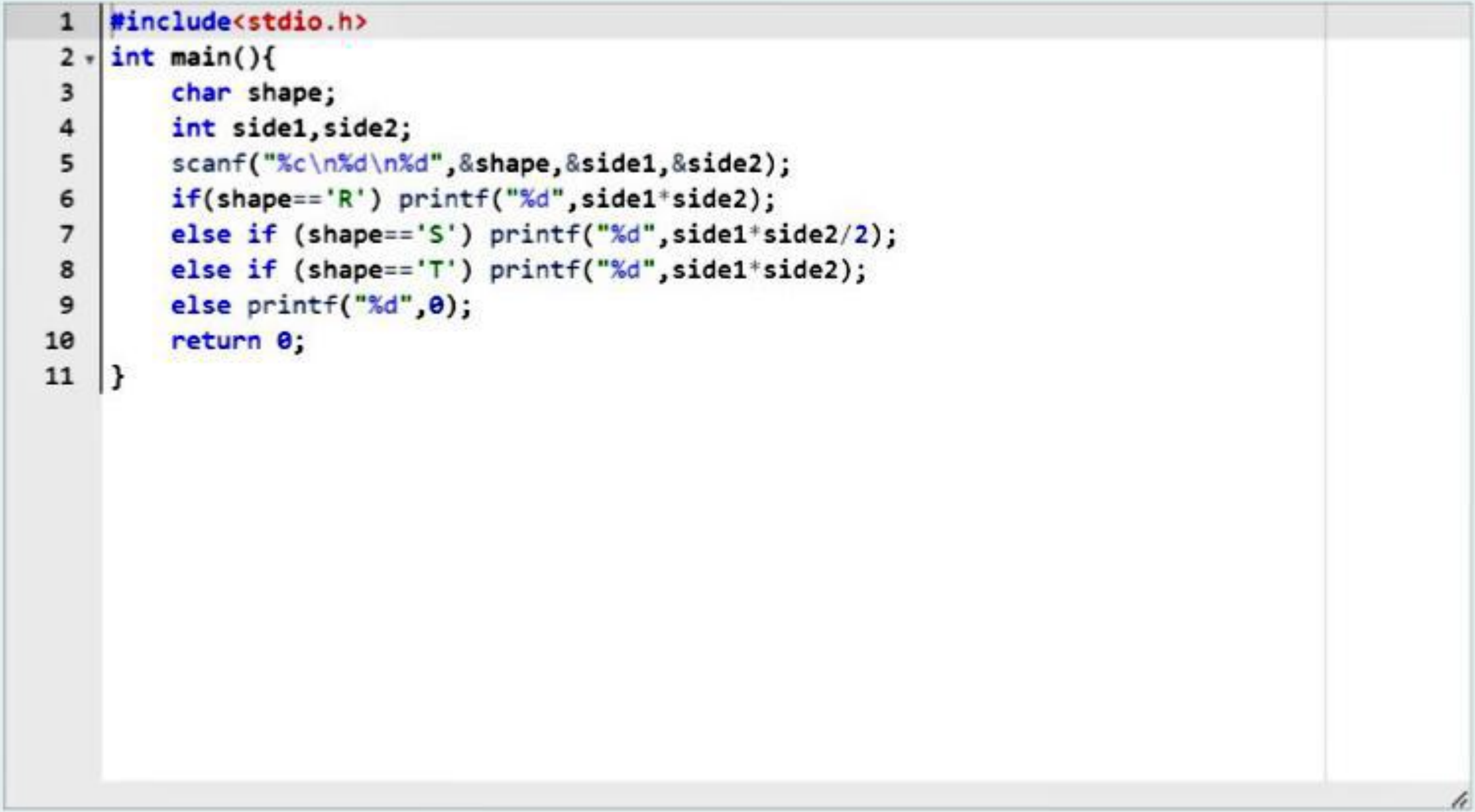
Output Format

- Print the area of the shape.

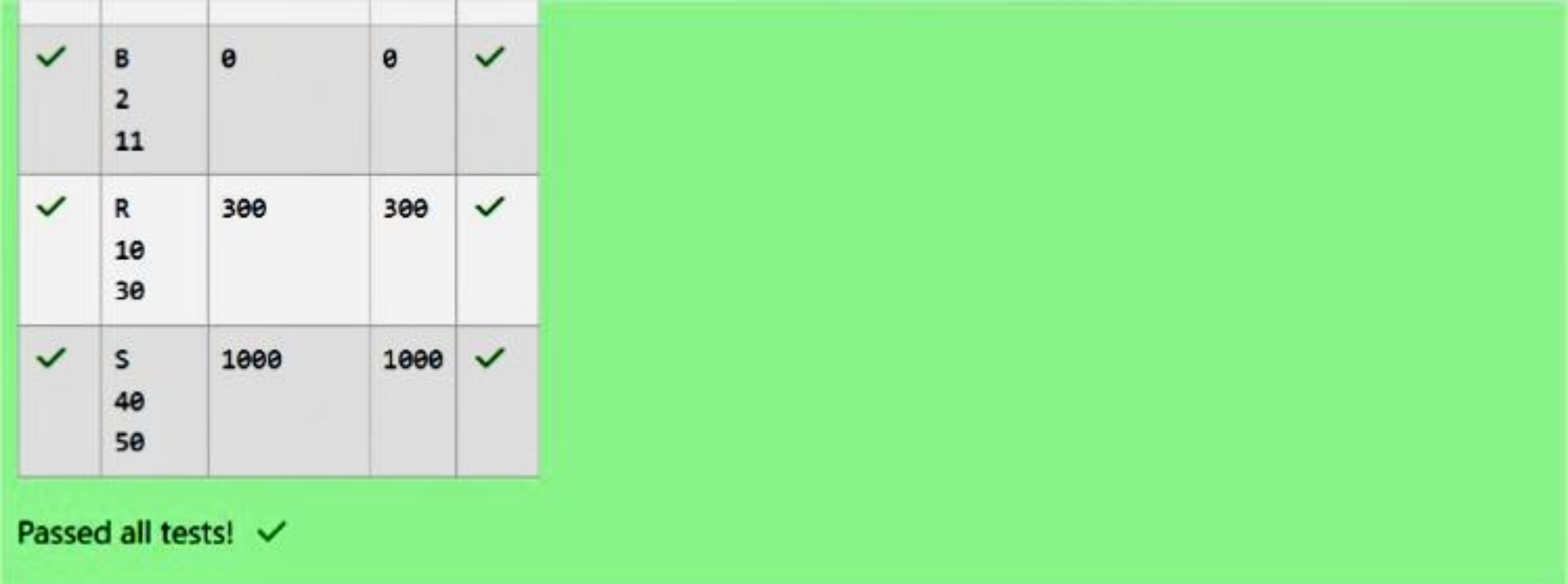
Sample Input 1

```
T
10
20
```

Sample Output 1

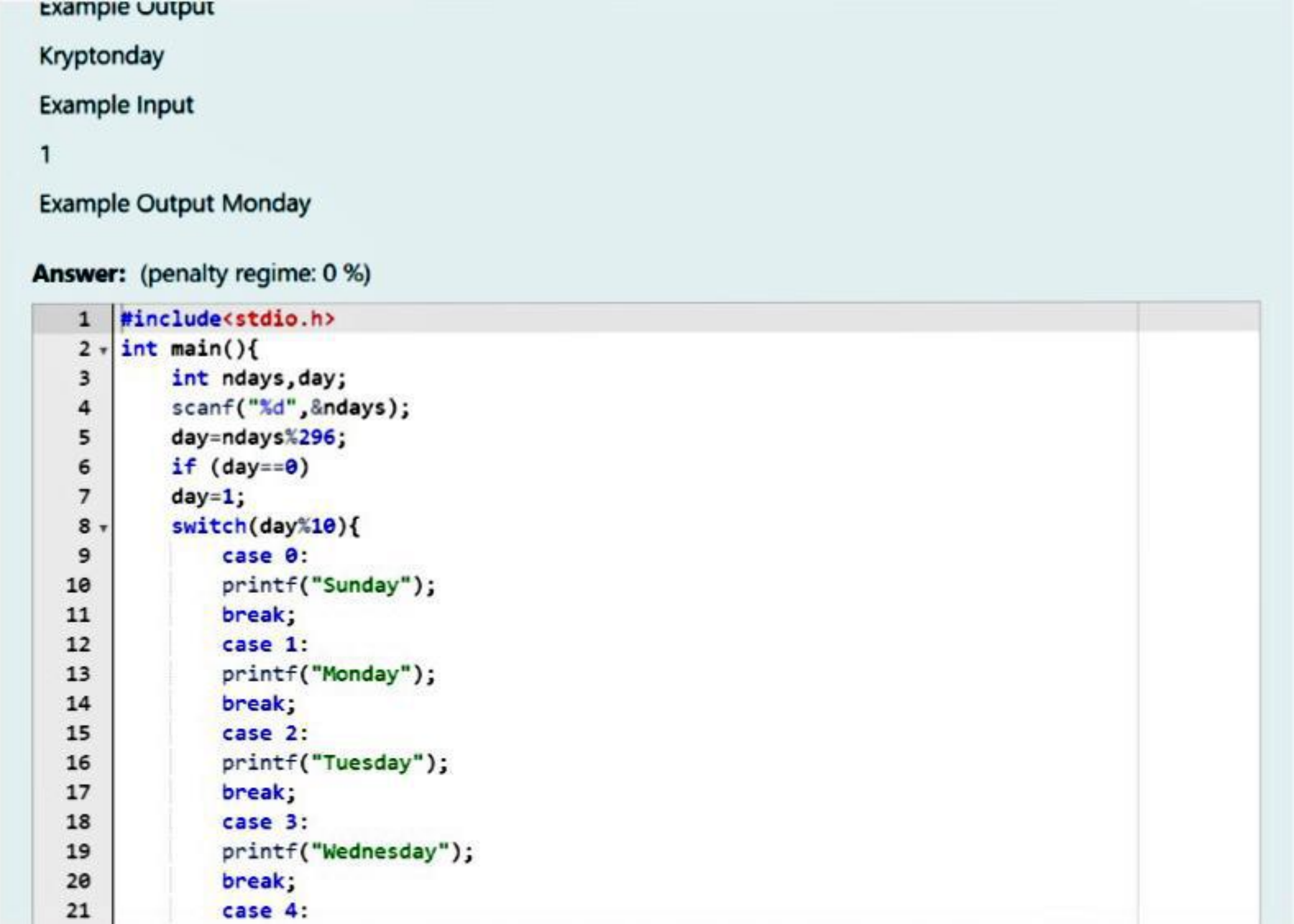


```c
#include<stdio.h>
int main(){
    char shape;
    int side1,side2;
    scanf("%c\n%d\n%d",&shape,&side1,&side2);
    if(shape=='R') printf("%d",side1*side2);
    else if (shape=='S') printf("%d",side1*side2/2);
    else if (shape=='T') printf("%d",side1*side2);
    else printf("%d",0);
    return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



✓	B 2 11	0	0	✓
✓	R 10 30	300	300	✓
✓	S 40 50	1000	1000	✓

Passed all tests! ✓



Example Output

Kryptonday

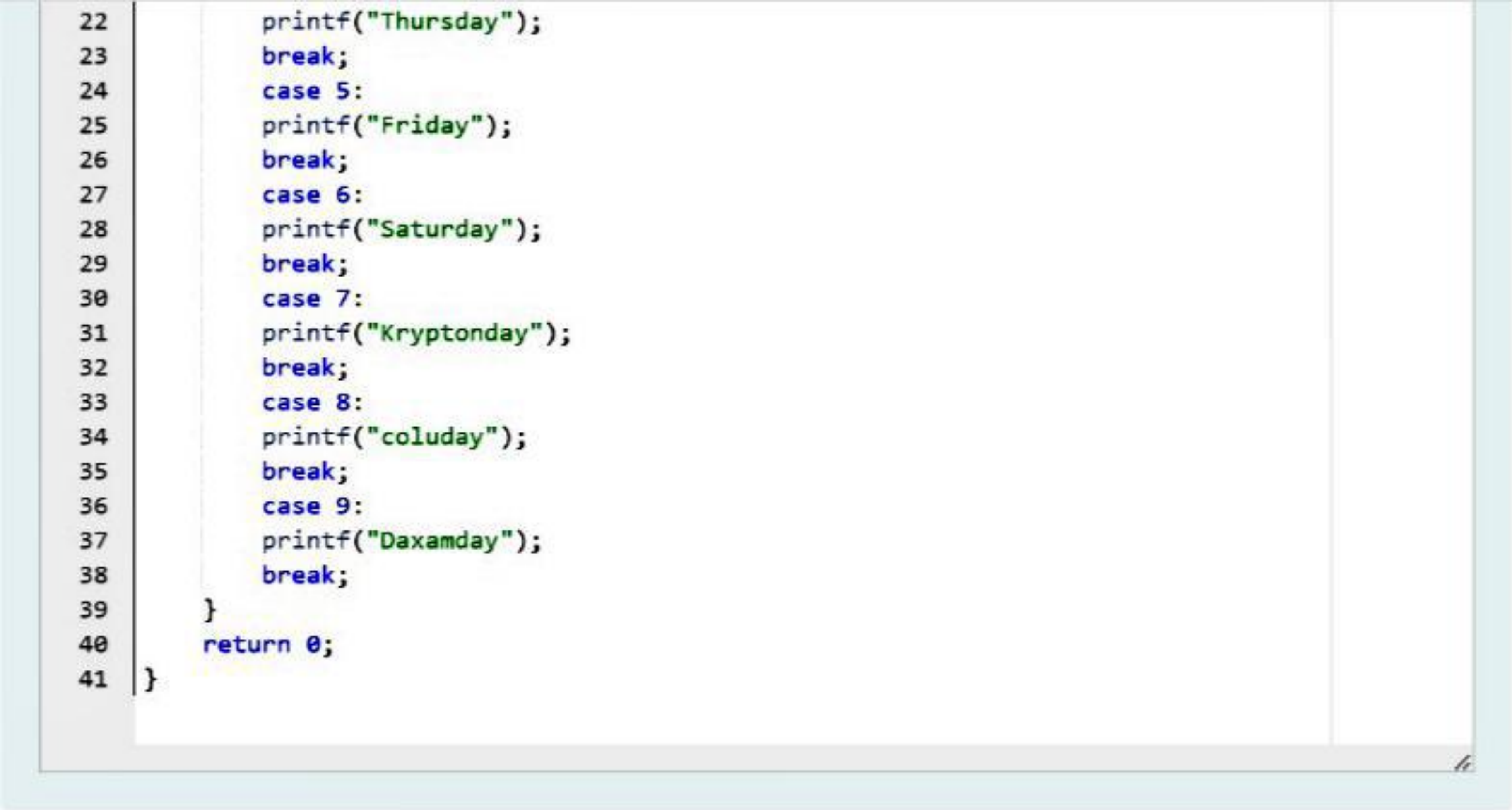
Example Input

1

Example Output Monday

Answer: (penalty regime: 0 %)

```
#include<stdio.h>
int main(){
    int ndays,day;
    scanf("%d",&ndays);
    day=ndays%296;
    if (day==0)
    day=1;
    switch(day%10){
        case 0:
        printf("Sunday");
        break;
        case 1:
        printf("Monday");
        break;
        case 2:
        printf("Tuesday");
        break;
        case 3:
        printf("Wednesday");
        break;
        case 4:
```



```
        printf("Thursday");
        break;
        case 5:
        printf("Friday");
        break;
        case 6:
        printf("Saturday");
        break;
        case 7:
        printf("Kryptonday");
        break;
        case 8:
        printf("coluday");
        break;
        case 9:
        printf("Daxamday");
        break;
    }
    return 0;
}
```