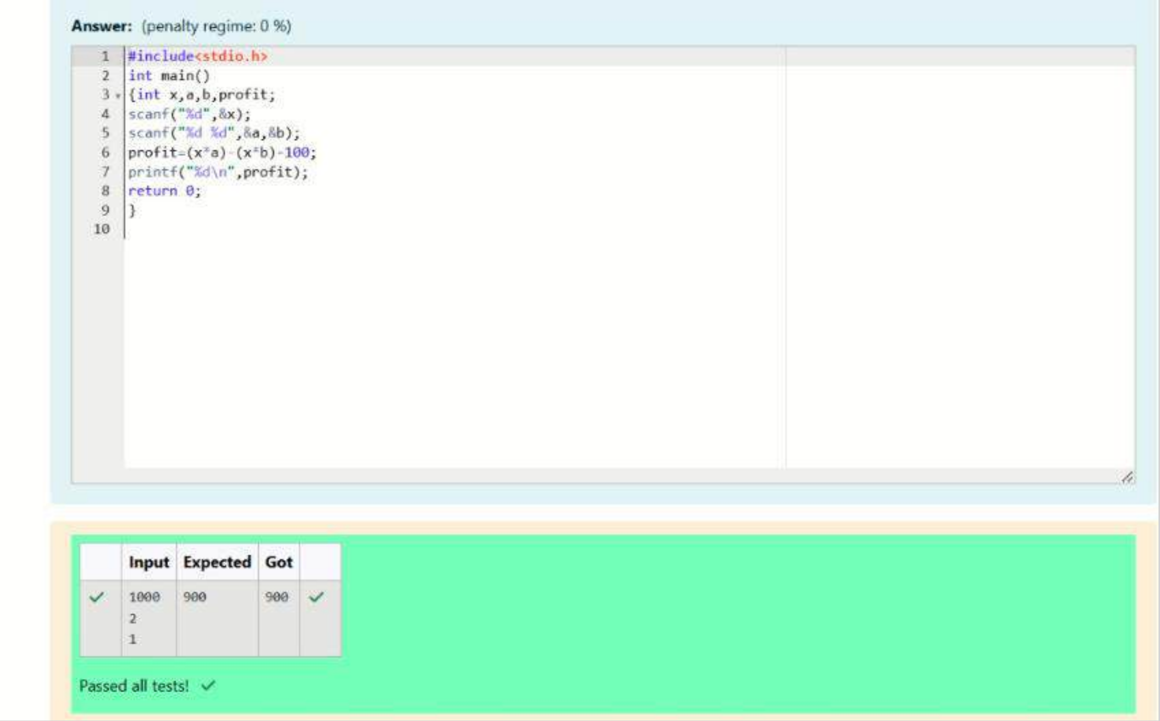
WEEK 2

Each Sunday, a newspaper agency sells X copies of a certain newspaper for Rs.A per copy. The cost to the agency of each newspaper is Rs.B . 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 obtained on Sundays. Can you please help them out by writing a C program to compute the profit given X, A and B.



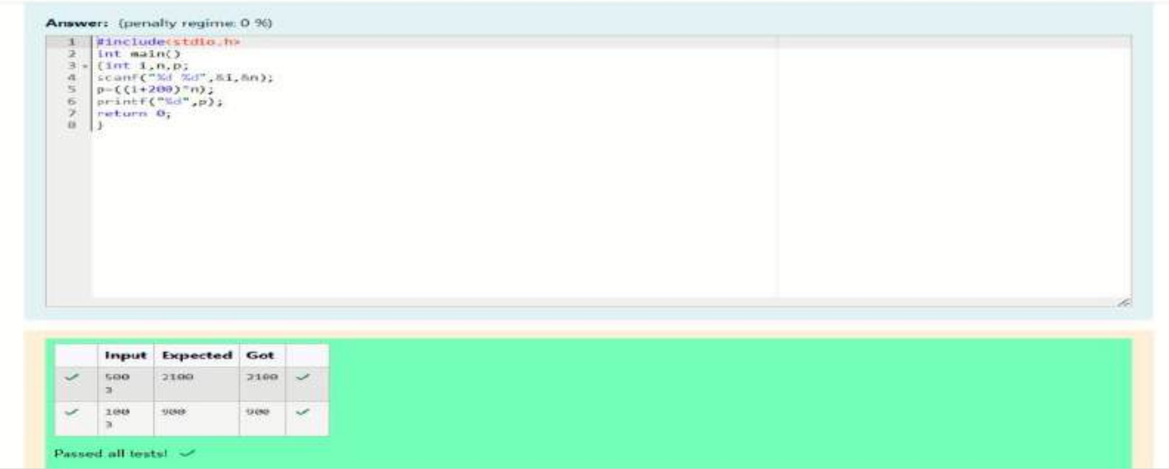
Answer: (penalty regime: 0 %)

```
#include<stdio.h>
int main()
{int x,a,b,profit;
scanf("%d",&x);
scanf("%d %d",&a,&b);
profit=(x*a)-(x*b)-100;
printf("%d\n",profit);
return 0;
}
```

	Input	Expected	Got	
✓	1000 2 1	900	900	✓

Passed all tests! ✓

Baba is very kind to beggars and every day Baba donates half of the amount he has when ever a beggar requests him. The money M left in Baba's hand is passed as the input and the number of beggars B who received the alms are passed as the input. The program must print the money Baba had in the beginning of the day.



Answer: (penalty regime: 0 %)

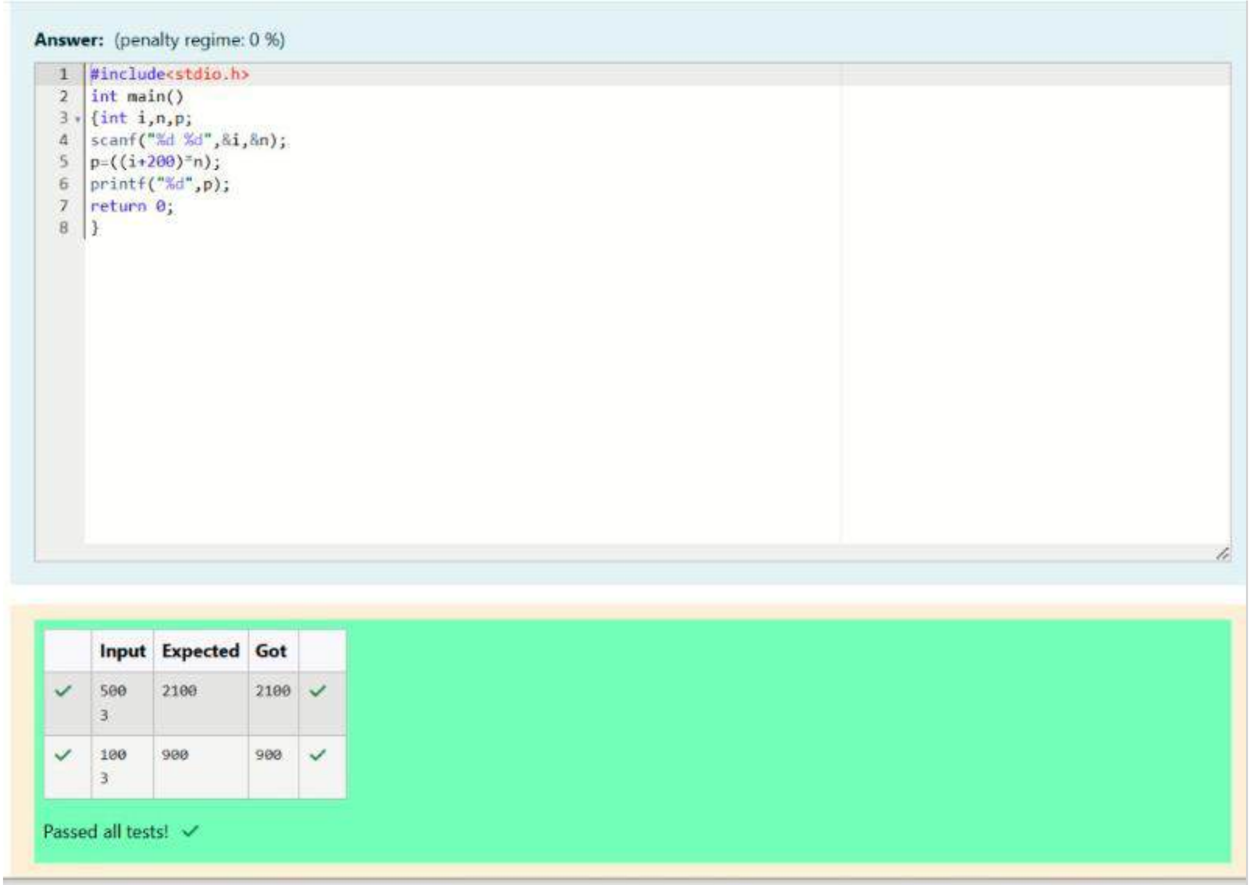
```
#include<stdio.h>
int main()
{int i,n,p;
scanf("%d %d",&i,&n);
p=((i+200)*n);
printf("%d",p);
return 0;
}
```

	Input	Expected	Got	
✓	500 3	2100	2100	✓
✓	100 3	900	900	✓

Passed all tests! ✓

The CEO of company ABC Inc wanted to encourage the employees coming on time to the office. So he announced that for every consecutive day an employee comes on time in a week (starting from

Monday to Saturday), he will be awarded Rs.200 more than the previous day as "Punctuality Incentive". The incentive I for the starting day (ie on Monday) is passed as the input to the program. The number of days N an employee came on time consecutively starting from Monday is also passed as the input. The program must calculate and print the "Punctuality Incentive" P of the employee.



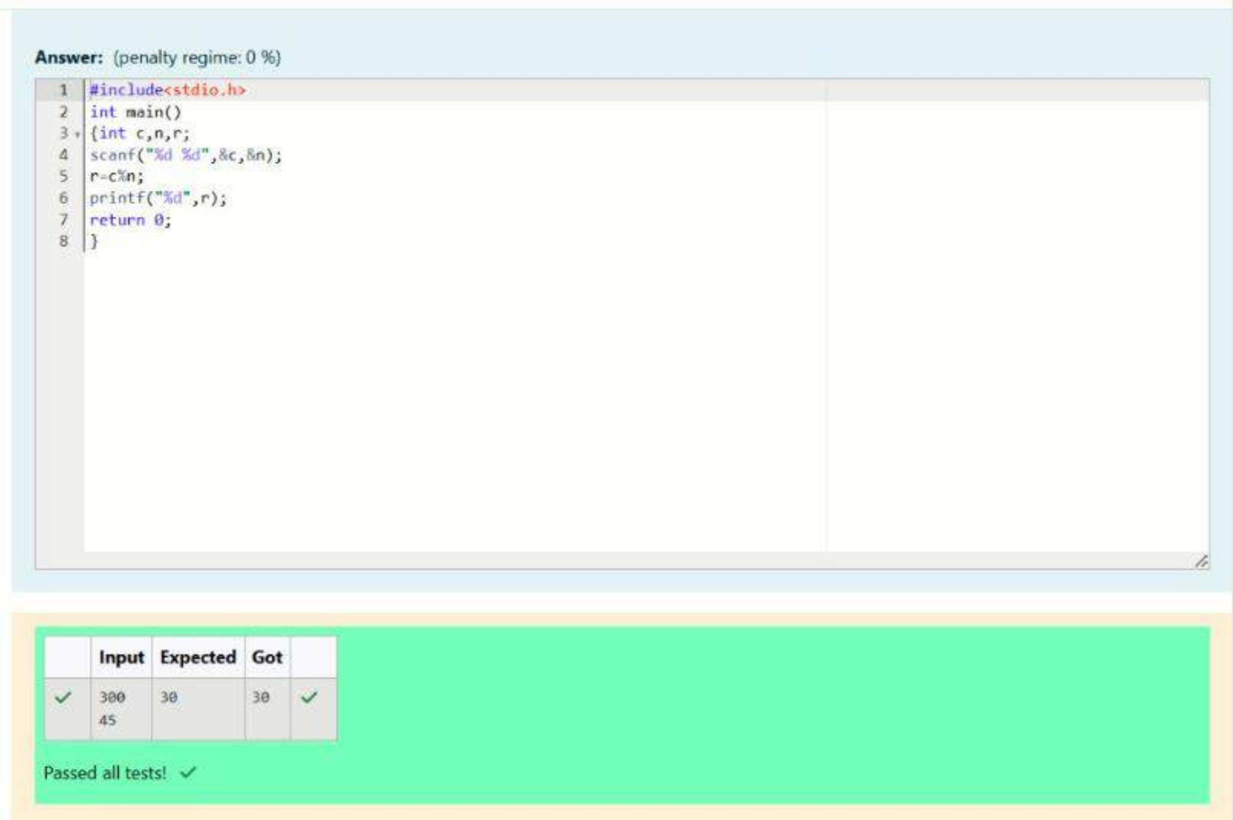
Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main()
{int i,n,p;
scanf("%d %d",&i,&n);
p=((i+200)*n);
printf("%d",p);
return 0;
}
```

	Input	Expected	Got	
✓	500 3	2100	2100	✓
✓	100 3	900	900	✓

Passed all tests! ✓

Bajan Lal distributes C chocolates to school N students every Friday. The C chocolates are distributed among N students equally and the remaining chocolates R are given back to Bajan Lal.As an example if C=100 and N=40, each student receives 2 chocolates and the balance 100-40*2 = 20 is given back.If C=205 and N=20, then each student receives 10 chocolates and the balance 205-20*10 = 5 is given back.Help the school to calculate the chocolates to be given back when C and N are passed as input



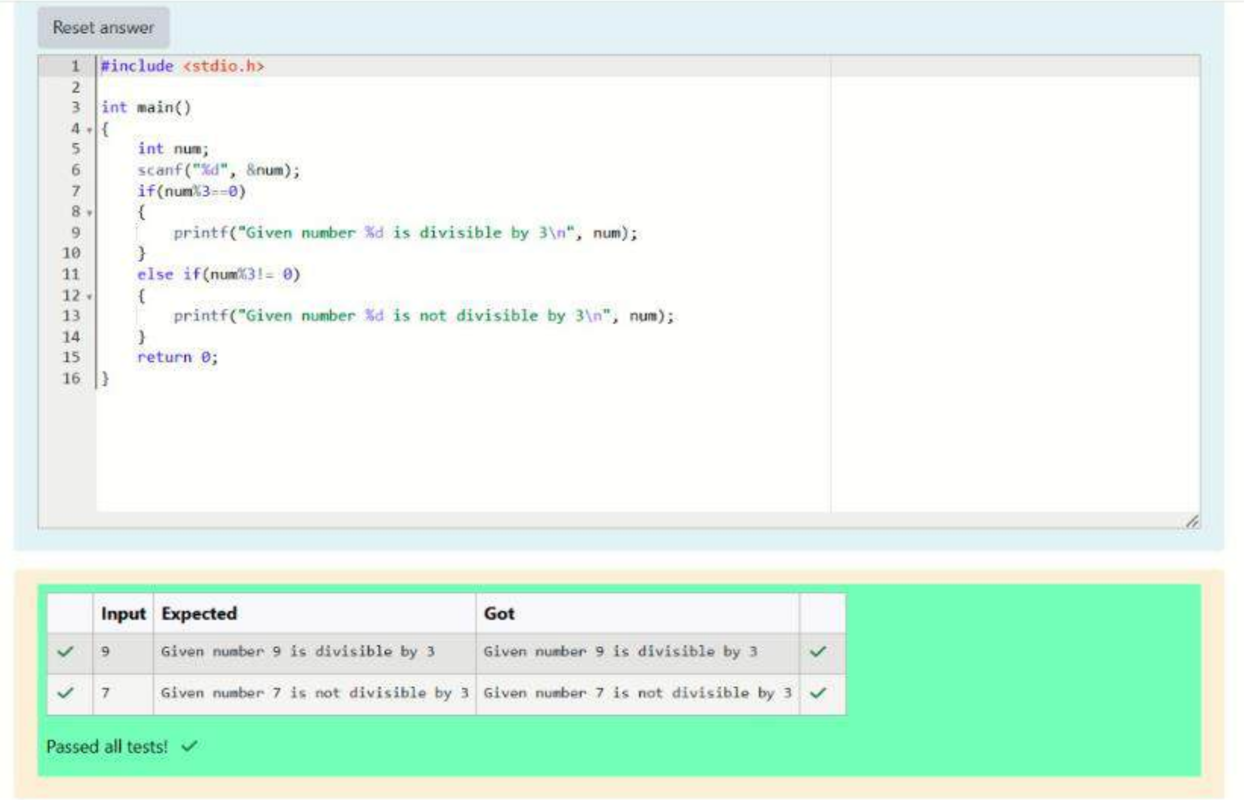
Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main()
{int c,n,r;
scanf("%d %d",&c,&n);
r=c%n;
printf("%d",r);
return 0;
}
```

	Input	Expected	Got	
✓	300 45	30	30	✓

Passed all tests! ✓

If construct

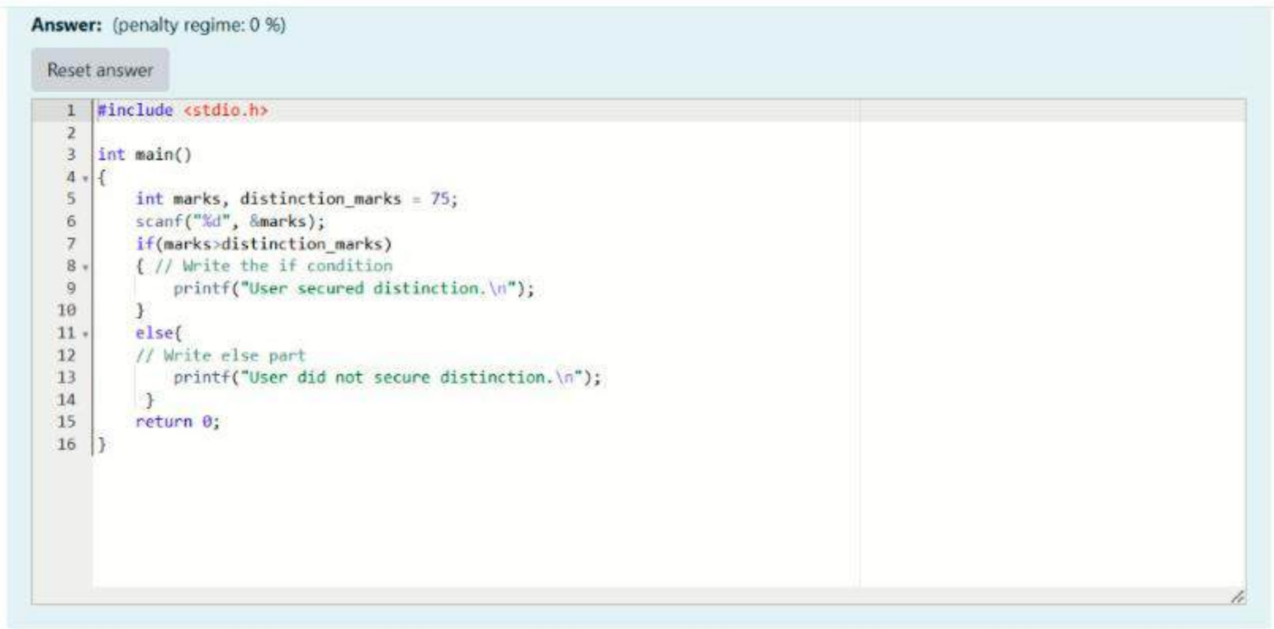


```c
#include <stdio.h>

int main()
{
    int num;
    scanf("%d", &num);
    if(num%3==0)
    {
        printf("Given number %d is divisible by 3\n", num);
    }
    else if(num%3!= 0)
    {
        printf("Given number %d is not divisible by 3\n", num);
    }
    return 0;
}
```

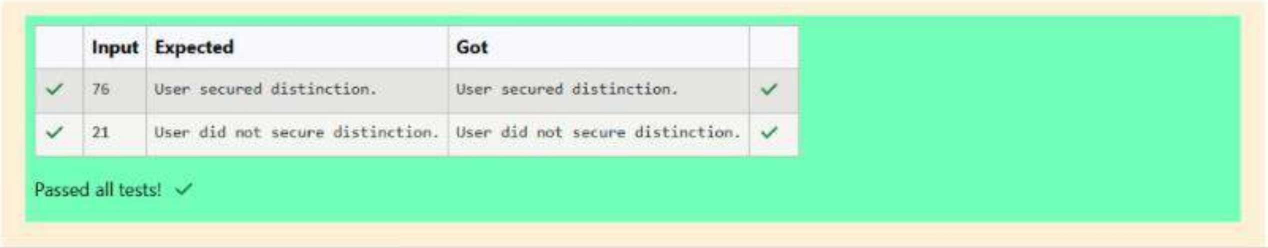
	Input	Expected	Got	
✓	9	Given number 9 is divisible by 3	Given number 9 is divisible by 3	✓
✓	7	Given number 7 is not divisible by 3	Given number 7 is not divisible by 3	✓

Passed all tests! ✓

If-else construct



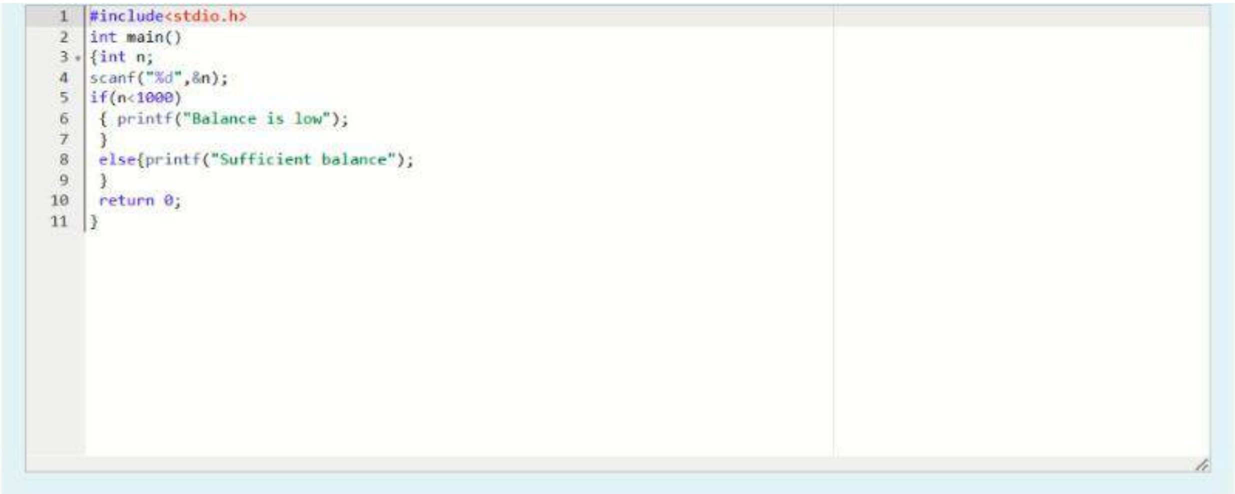
```c
#include <stdio.h>

int main()
{
    int marks, distinction_marks = 75;
    scanf("%d", &marks);
    if(marks>distinction_marks)
    { // Write the if condition
        printf("User secured distinction.\n");
    }
    else{
    // Write else part
        printf("User did not secure distinction.\n");
     }
    return 0;
}
```



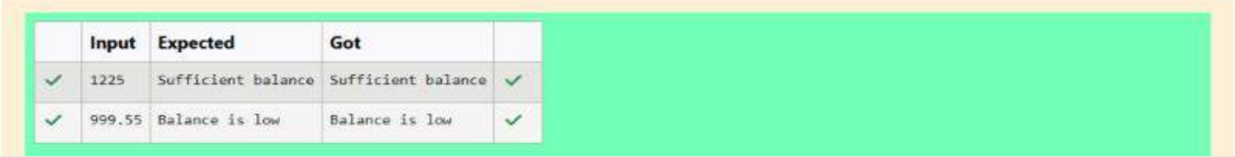
	Input	Expected	Got	
✓	76	User secured distinction.	User secured distinction.	✓
✓	21	User did not secure distinction.	User did not secure distinction.	✓

Passed all tests! ✓

Write code which uses an if-else statement to check whether a given account balance is greater or lesser than the minimum balance.

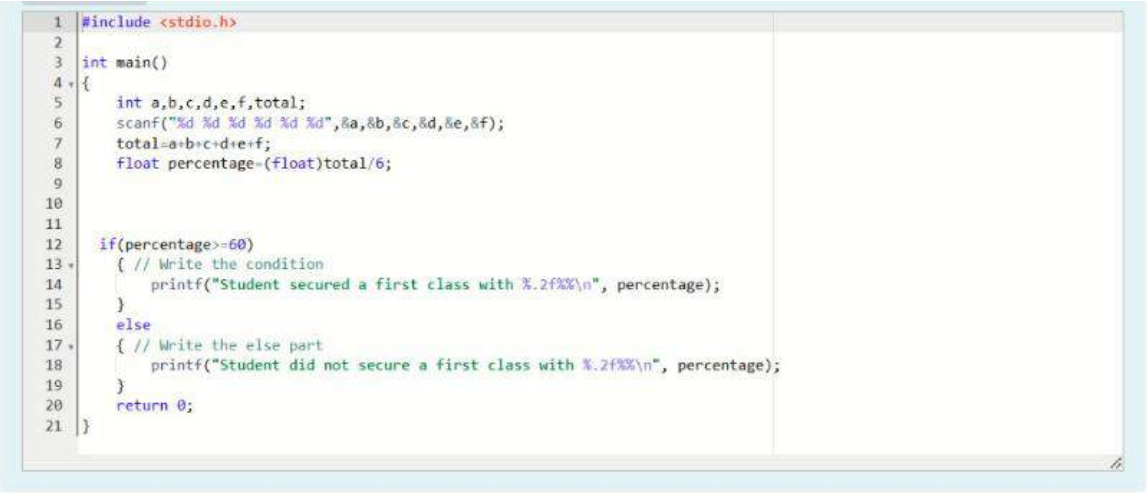


```c
#include<stdio.h>
int main()
{int n;
scanf("%d",&n);
if(n<1000)
 { printf("Balance is low");
 }
 else{printf("Sufficient balance");
 }
 return 0;
}
```

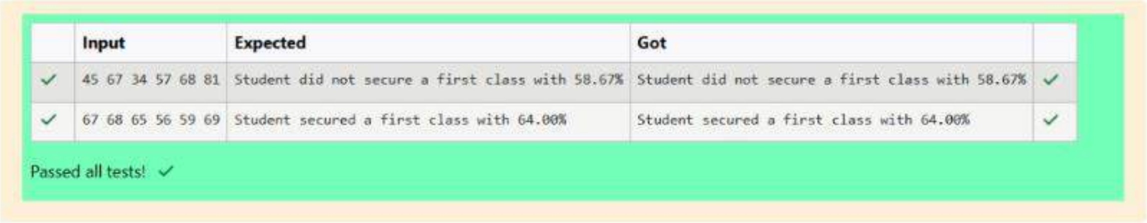


	Input	Expected	Got	
✓	1225	Sufficient balance	Sufficient balance	✓
✓	999.55	Balance is low	Balance is low	✓

Fill in the missing code in the below program to check whether the student secured first class or not.



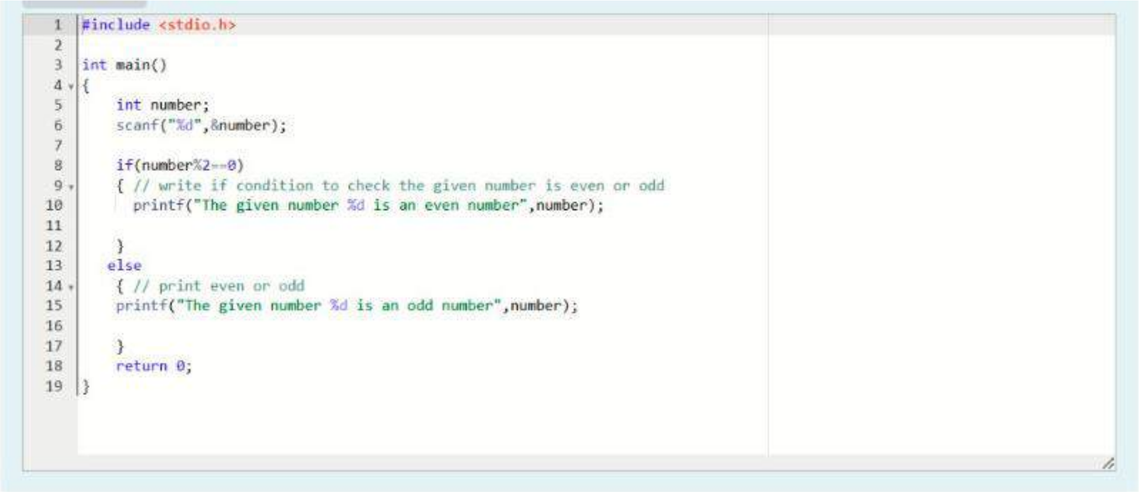
```c
#include <stdio.h>

int main()
{
    int a,b,c,d,e,f,total;
    scanf("%d %d %d %d %d %d",&a,&b,&c,&d,&e,&f);
    total=a+b+c+d+e+f;
    float percentage=(float)total/6;


  if(percentage>=60)
    { // Write the condition
        printf("Student secured a first class with %.2f%%\n", percentage);
    }
    else
    { // Write the else part
        printf("Student did not secure a first class with %.2f%%\n", percentage);
    }
    return 0;
}
```



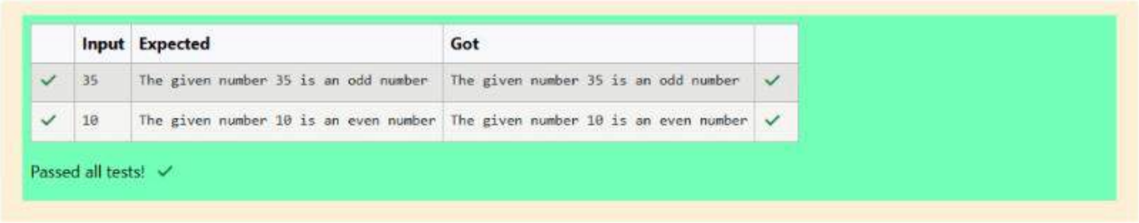
	Input	Expected	Got	
✓	45 67 34 57 68 81	Student did not secure a first class with 58.67%	Student did not secure a first class with 58.67%	✓
✓	67 68 65 56 59 69	Student secured a first class with 64.00%	Student secured a first class with 64.00%	✓

Passed all tests! ✓

Write a program which uses an if-else statement to verify and print if the given number is an odd or an even



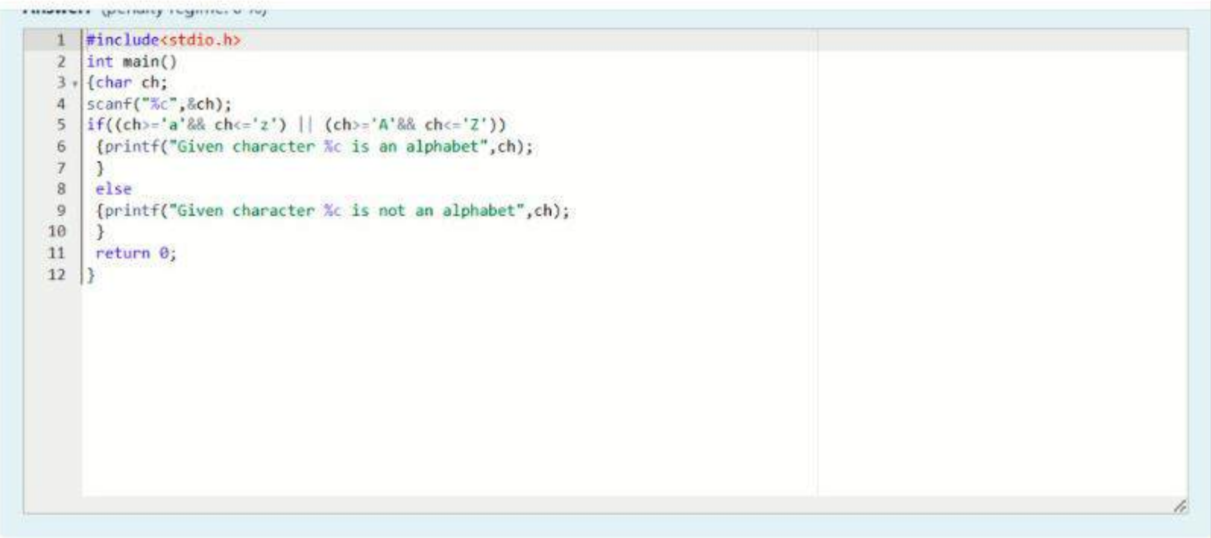
```c
#include <stdio.h>

int main()
{
    int number;
    scanf("%d",&number);

    if(number%2==0)
    { // write if condition to check the given number is even or odd
      printf("The given number %d is an even number",number);

    }
   else
    { // print even or odd
    printf("The given number %d is an odd number",number);

    }
    return 0;
}
```



	Input	Expected	Got	
✓	35	The given number 35 is an odd number	The given number 35 is an odd number	✓
✓	10	The given number 10 is an even number	The given number 10 is an even number	✓

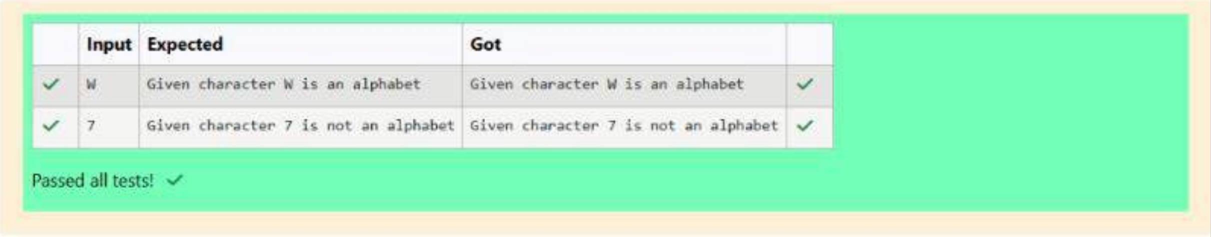
Passed all tests! ✓

Write a program which uses an if-else statement to verify if the given character is an alphabet or not.



Answer: (penalty regime: 0 %)

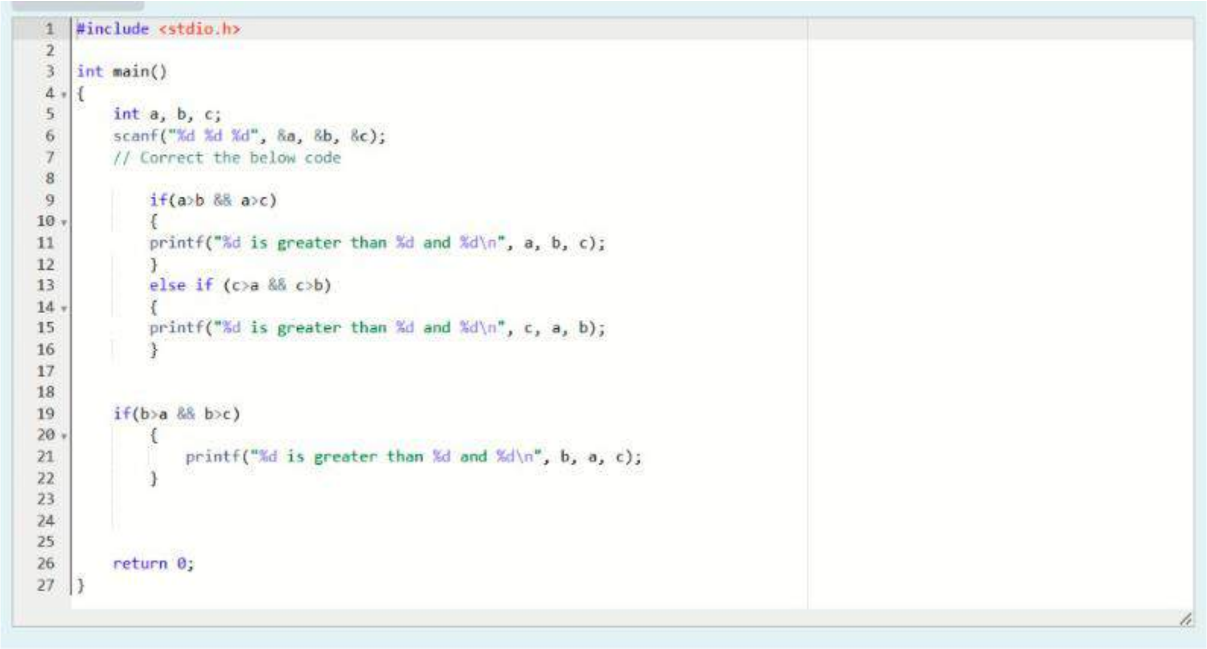
```
#include<stdio.h>
int main()
{char ch;
scanf("%c",&ch);
if((ch>='a'&& ch<='z') || (ch>='A'&& ch<='Z'))
 {printf("Given character %c is an alphabet",ch);
 }
 else
 {printf("Given character %c is not an alphabet",ch);
 }
 return 0;
}
```



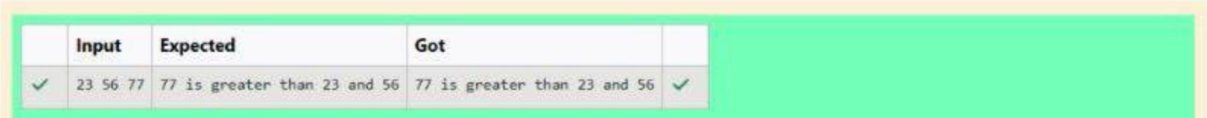
	Input	Expected	Got	
✓	W	Given character W is an alphabet	Given character W is an alphabet	✓
✓	7	Given character 7 is not an alphabet	Given character 7 is not an alphabet	✓

Passed all tests! ✓

nested if-else construct:

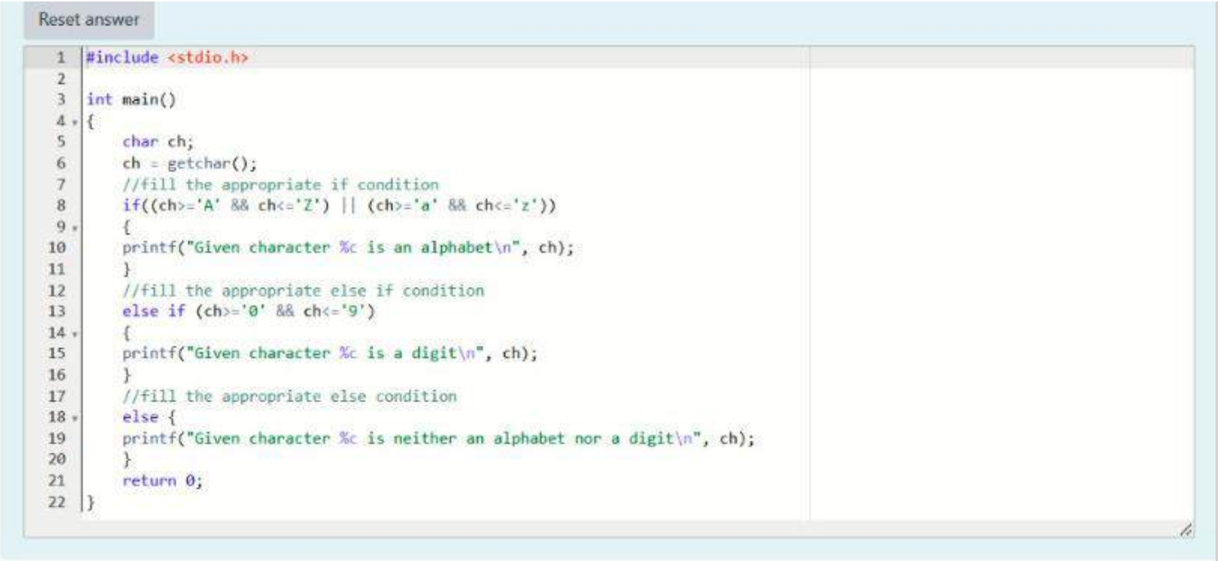


```
#include <stdio.h>

int main()
{
    int a, b, c;
    scanf("%d %d %d", &a, &b, &c);
    // Correct the below code

        if(a>b && a>c)
        {
        printf("%d is greater than %d and %d\n", a, b, c);
        }
        else if (c>a && c>b)
        {
        printf("%d is greater than %d and %d\n", c, a, b);
        }


    if(b>a && b>c)
        {
            printf("%d is greater than %d and %d\n", b, a, c);
        }



    return 0;
}
```

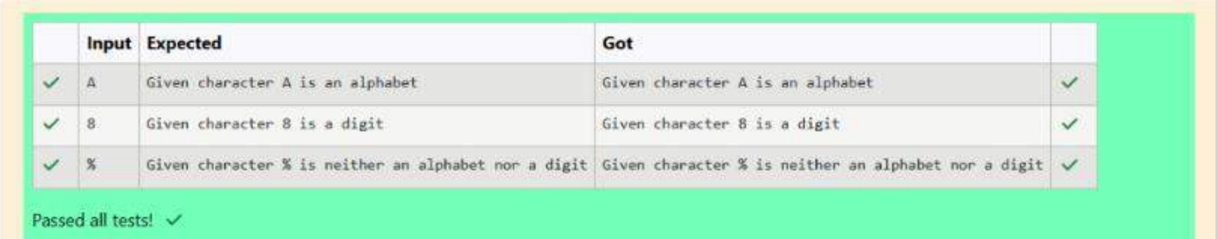


	Input	Expected	Got	
✓	23 56 77	77 is greater than 23 and 56	77 is greater than 23 and 56	✓

If-else-if construct



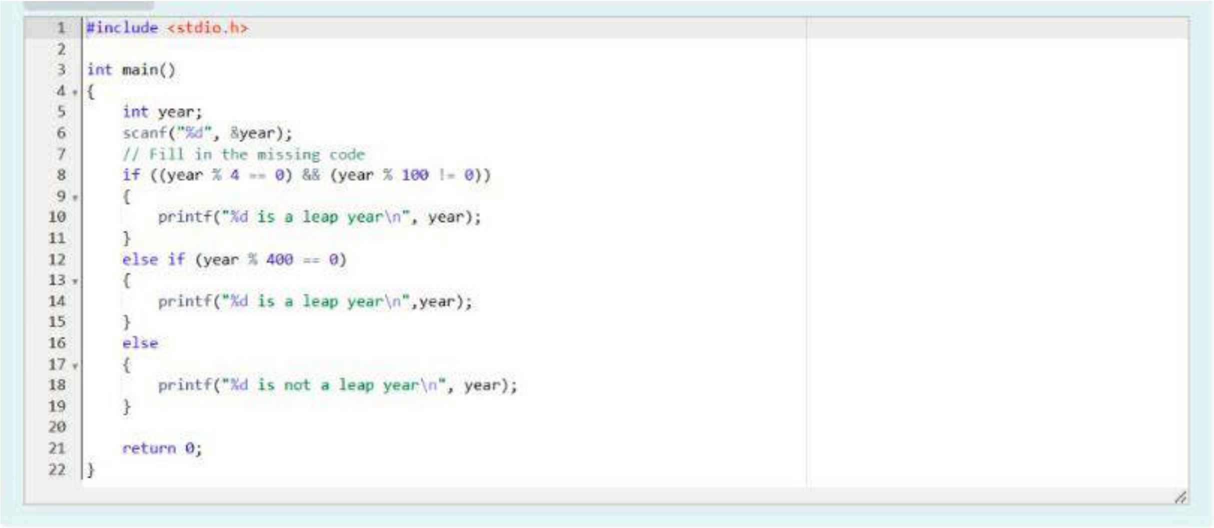
```c
#include <stdio.h>

int main()
{
    char ch;
    ch = getchar();
    //fill the appropriate if condition
    if((ch>='A' && ch<='Z') || (ch>='a' && ch<='z'))
    {
    printf("Given character %c is an alphabet\n", ch);
    }
    //fill the appropriate else if condition
    else if (ch>='0' && ch<='9')
    {
    printf("Given character %c is a digit\n", ch);
    }
    //fill the appropriate else condition
    else {
    printf("Given character %c is neither an alphabet nor a digit\n", ch);
    }
    return 0;
}
```



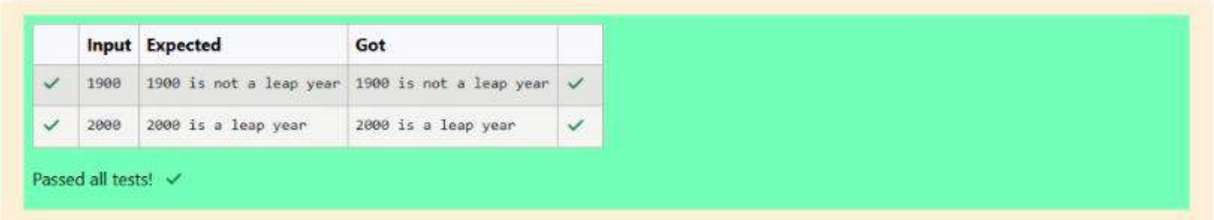
	Input	Expected	Got	
✓	A	Given character A is an alphabet	Given character A is an alphabet	✓
✓	8	Given character 8 is a digit	Given character 8 is a digit	✓
✓	%	Given character % is neither an alphabet nor a digit	Given character % is neither an alphabet nor a digit	✓

Passed all tests! ✓

The following code uses if-else statement to check whether the given integer number is a valid **leap year** or not.



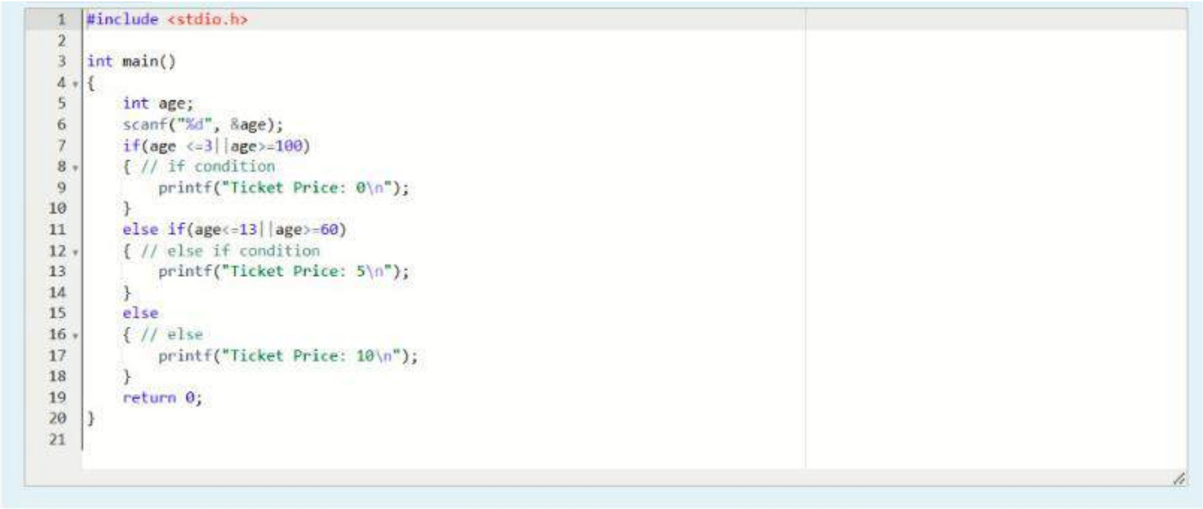
```c
#include <stdio.h>

int main()
{
    int year;
    scanf("%d", &year);
    // Fill in the missing code
    if ((year % 4 == 0) && (year % 100 != 0))
    {
        printf("%d is a leap year\n", year);
    }
    else if (year % 400 == 0)
    {
        printf("%d is a leap year\n",year);
    }
    else
    {
        printf("%d is not a leap year\n", year);
    }

    return 0;
}
```



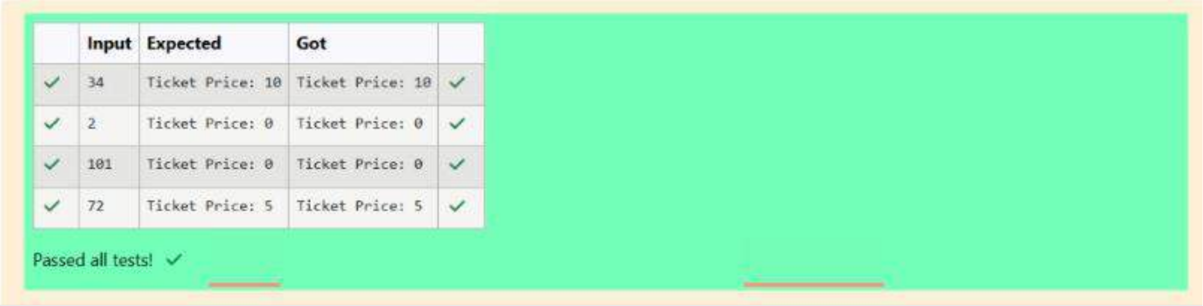
	Input	Expected	Got	
✓	1900	1900 is not a leap year	1900 is not a leap year	✓
✓	2000	2000 is a leap year	2000 is a leap year	✓

Passed all tests! ✓

Fill in the missing code in the below program to read an **integer value** for a variable age and use if-else statement to check the age and print appropriate ticket price.



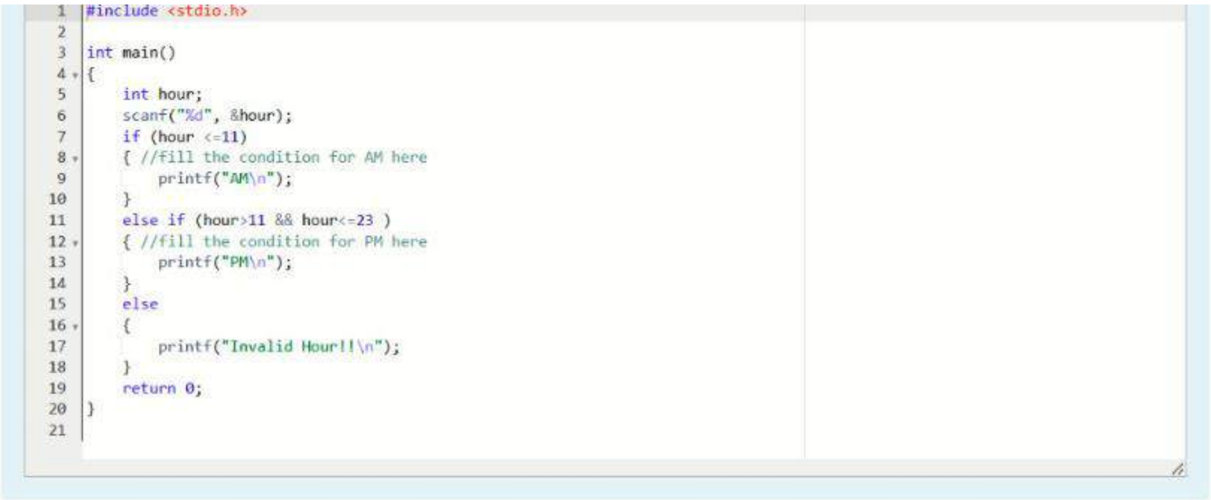
```c
#include <stdio.h>

int main()
{
    int age;
    scanf("%d", &age);
    if(age <=3||age>=100)
    { // if condition
        printf("Ticket Price: 0\n");
    }
    else if(age<=13||age>=60)
    { // else if condition
        printf("Ticket Price: 5\n");
    }
    else
    { // else
        printf("Ticket Price: 10\n");
    }
    return 0;
}
```



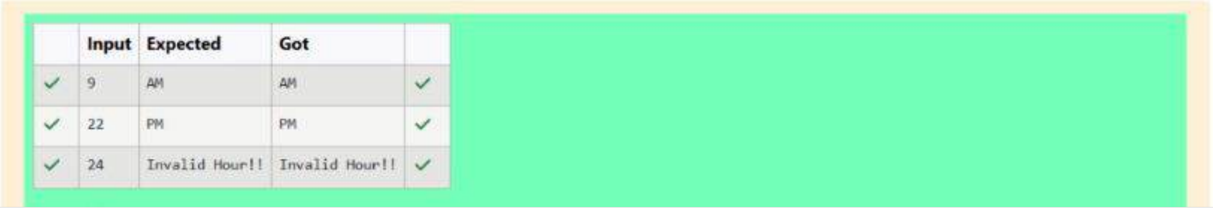
	Input	Expected	Got	
✓	34	Ticket Price: 10	Ticket Price: 10	✓
✓	2	Ticket Price: 0	Ticket Price: 0	✓
✓	101	Ticket Price: 0	Ticket Price: 0	✓
✓	72	Ticket Price: 5	Ticket Price: 5	✓

Passed all tests! ✓

See the below code which uses a if-else-if statement for calculating **AM** or **PM** for a given **hour**.



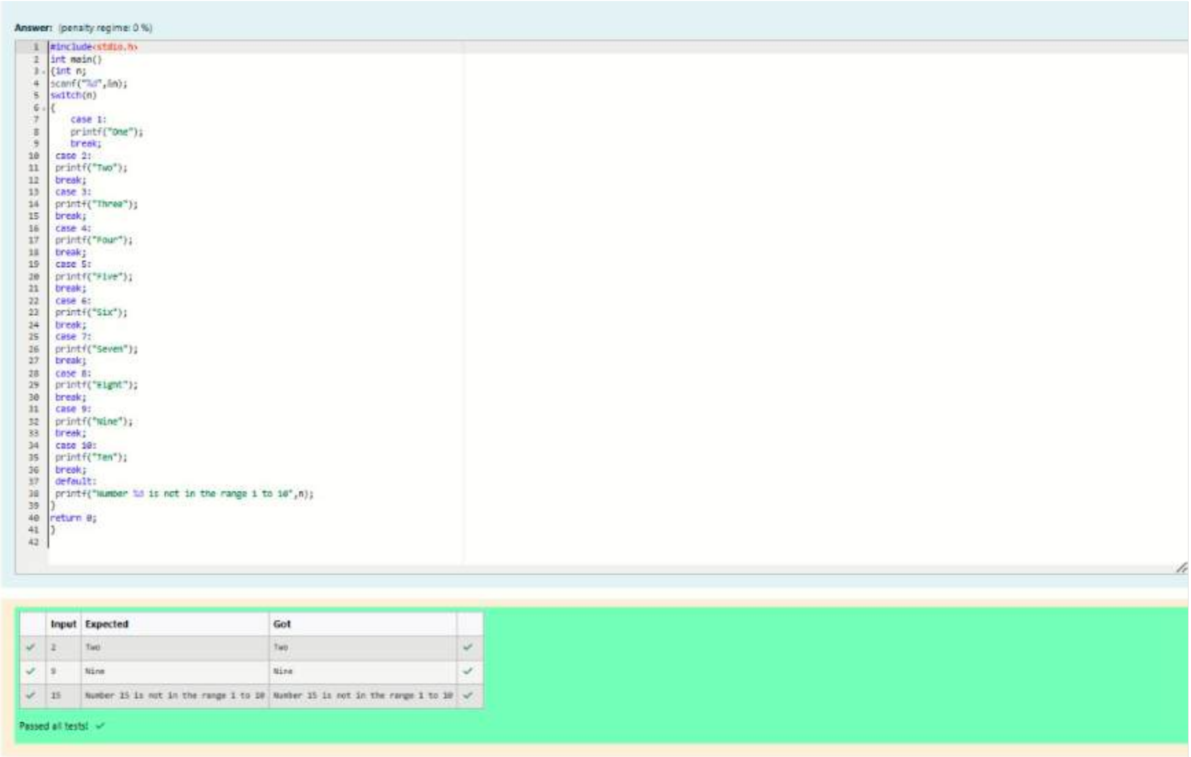
```c
#include <stdio.h>

int main()
{
    int hour;
    scanf("%d", &hour);
    if (hour <=11)
    { //fill the condition for AM here
        printf("AM\n");
    }
    else if (hour>11 && hour<=23 )
    { //fill the condition for PM here
        printf("PM\n");
    }
    else
    {
        printf("Invalid Hour!!\n");
    }
    return 0;
}
```



	Input	Expected	Got	
✓	9	AM	AM	✓
✓	22	PM	PM	✓
✓	24	Invalid Hour!!	Invalid Hour!!	✓

SWITCH CASE

A switch statement is used to change the control flow of a program execution through multiple paths depending on an expression's value



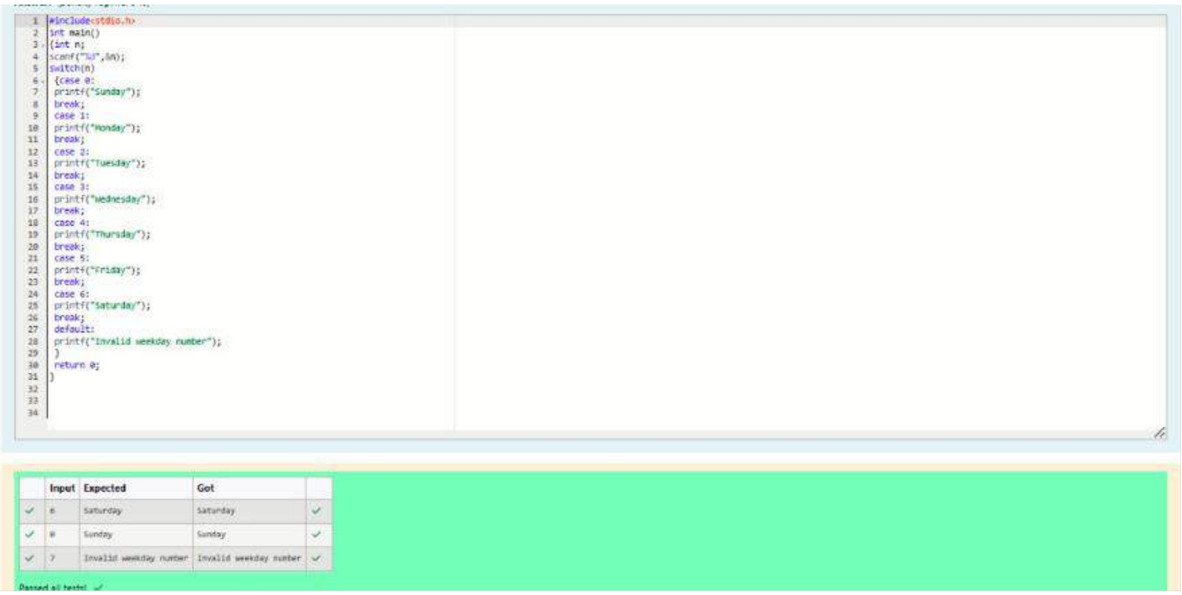
Answer: (penalty regime: 0 %)

```
#include<stdio.h>
int main()
{int n;
scanf("%d",&n);
switch(n)
{
    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;
 case 10:
 printf("Ten");
 break;
 default:
 printf("Number %d is not in the range 1 to 10",n);
}
return 0;
}
```

	Input	Expected	Got	
✓	2	Two	Two	✓
✓	9	Nine	Nine	✓
✓	15	Number 15 is not in the range 1 to 10	Number 15 is not in the range 1 to 10	✓

Passed all tests! ✓

Write a program to read the **weekday number** from the standard input and print the **weekday name** using switch-case.

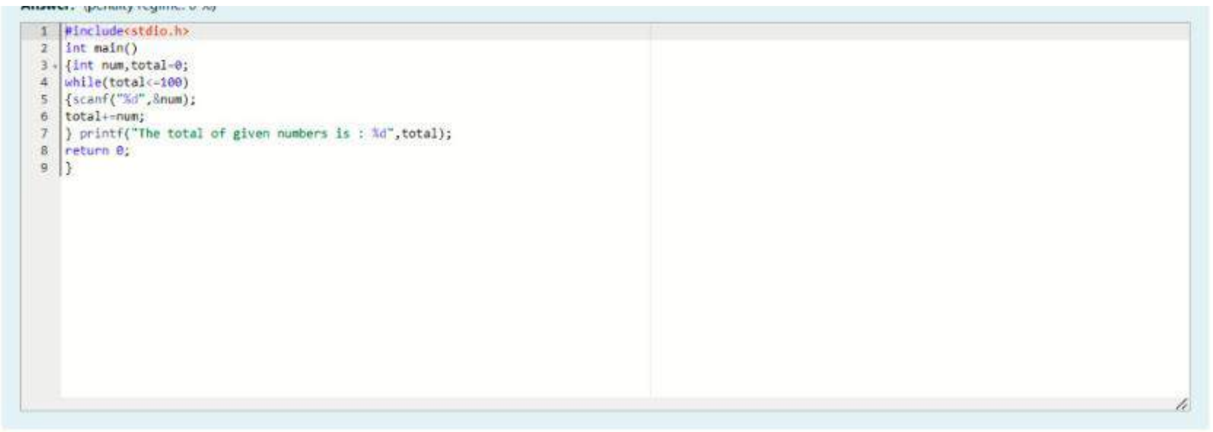


```
#include<stdio.h>
int main()
{int n;
scanf("%d",&n);
switch(n)
 {case 0:
 printf("Sunday");
 break;
 case 1:
 printf("Monday");
 break;
 case 2:
 printf("Tuesday");
 break;
 case 3:
 printf("Wednesday");
 break;
 case 4:
 printf("Thursday");
 break;
 case 5:
 printf("Friday");
 break;
 case 6:
 printf("Saturday");
 break;
 default:
 printf("Invalid weekday number");
 }
 return 0;
}
```

	Input	Expected	Got	
✓	6	Saturday	Saturday	✓
✓	0	Sunday	Sunday	✓
✓	7	Invalid weekday number	Invalid weekday number	✓

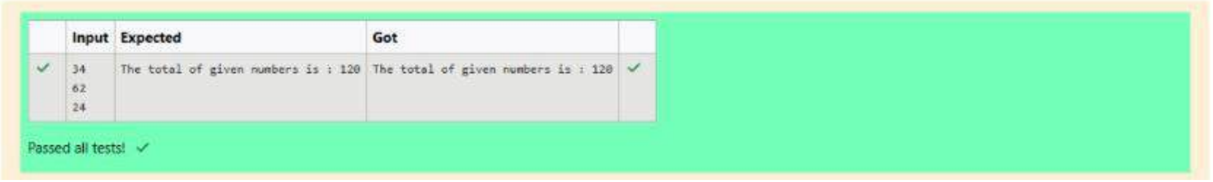
Passed all tests! ✓

Most of the programming languages provide a special construct/statement using which we can repeatedly execute one or more statement as long as a condition is **true**. In C, we have while, do-while and for as the three main looping constructs or statements.



Answer: (penalty regime: 0 %)

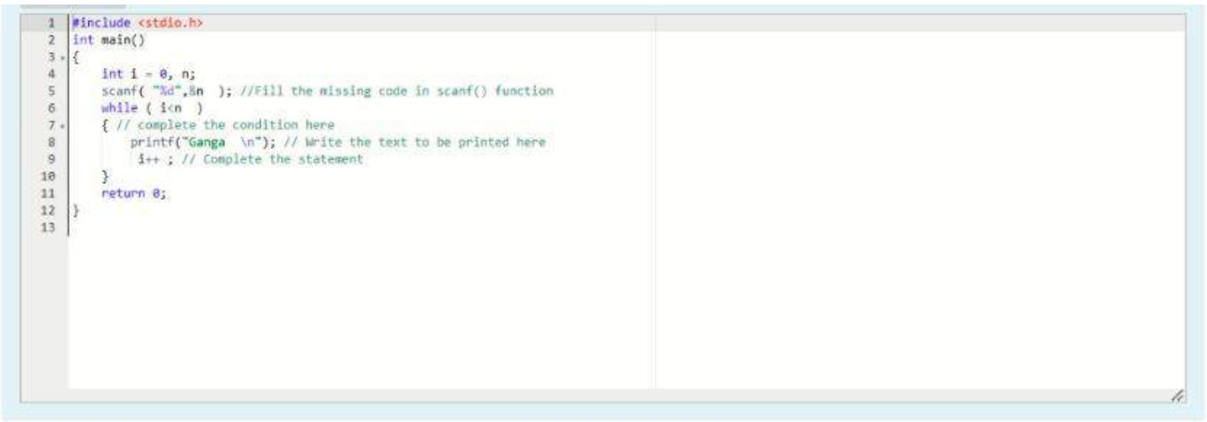
```
#include<stdio.h>
int main()
{int num,total=0;
while(total<=100)
{scanf("%d",&num);
total+=num;
} printf("The total of given numbers is : %d",total);
return 0;
}
```



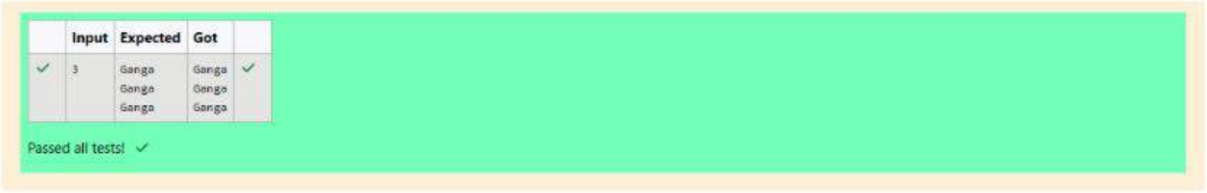
	Input	Expected	Got	
✓	34 62 24	The total of given numbers is : 120	The total of given numbers is : 120	✓

Passed all tests! ✓

The below sample code should print Ganga by number of times, where as the input is read by the programmer using **scanf()**.



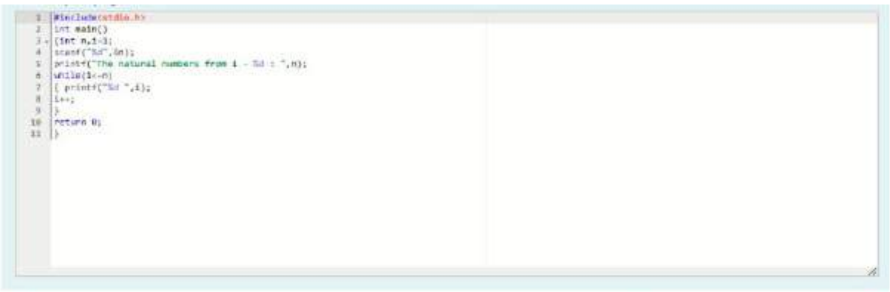
```
#include <stdio.h>
int main()
{
    int i = 0, n;
    scanf( "%d",&n  ); //Fill the missing code in scanf() function
    while ( i<n  )
    { // complete the condition here
        printf("Ganga  \n"); // Write the text to be printed here
         i++ ; // Complete the statement
    }
    return 0;
}
```



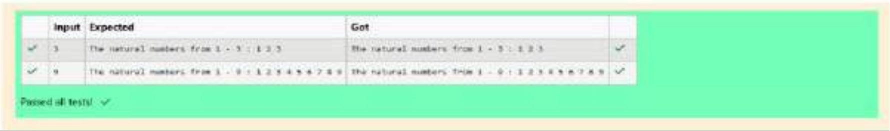
	Input	Expected	Got	
✓	3	Ganga Ganga Ganga	Ganga Ganga Ganga	✓

Passed all tests! ✓

Write a **C** program to print first n **natural numbers**.



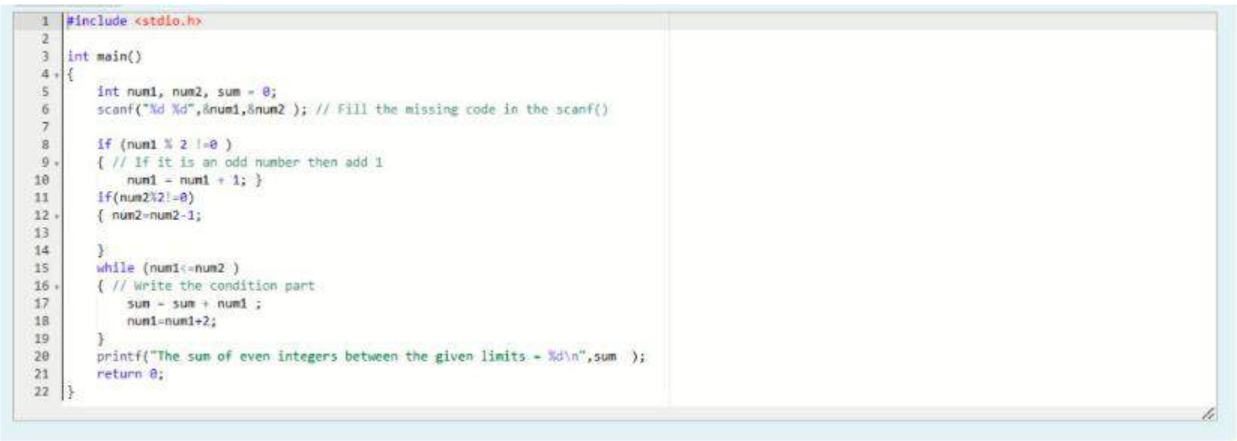
```
#include<stdio.h>
int main()
{int n,i=1;
scanf("%d",&n);
printf("The natural numbers from 1 - %d : ",n);
while(i<=n)
{ printf("%d ",i);
i++;
}
return 0;
}
```



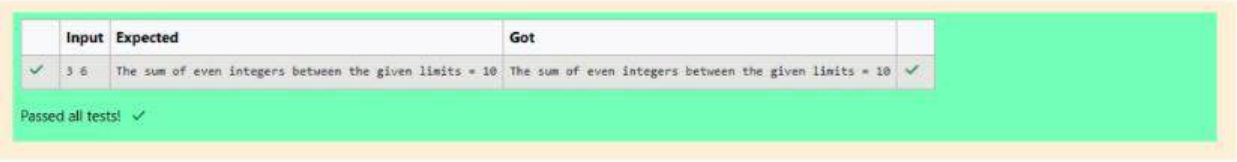
	Input	Expected	Got	
✓	3	The natural numbers from 1 - 3 : 1 2 3	The natural numbers from 1 - 3 : 1 2 3	✓
✓	9	The natural numbers from 1 - 9 : 1 2 3 4 5 6 7 8 9	The natural numbers from 1 - 9 : 1 2 3 4 5 6 7 8 9	✓

Passed all tests! ✓

The below sample code should find the sum of **even numbers** between any two numbers.



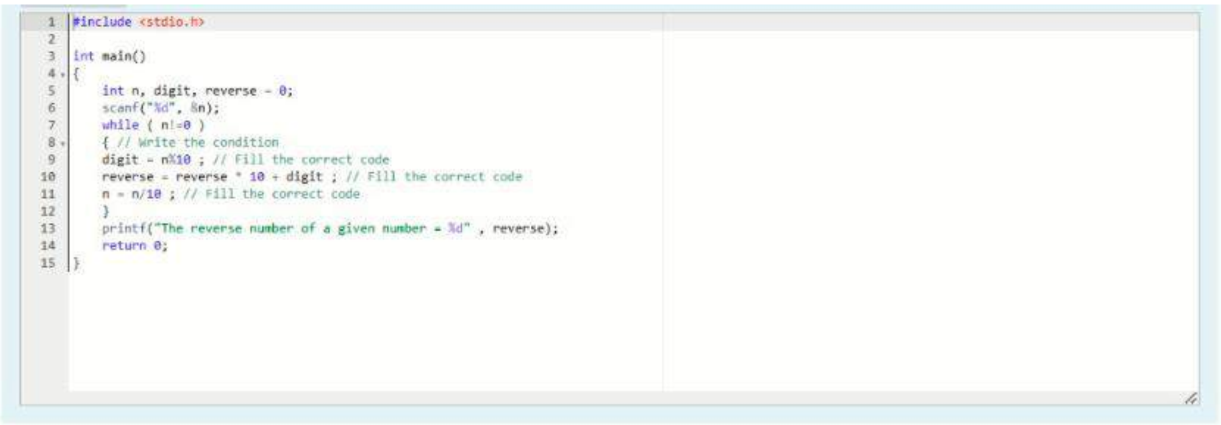
```
#include <stdio.h>

int main()
{
    int num1, num2, sum = 0;
    scanf("%d %d",&num1,&num2 ); // Fill the missing code in the scanf()

    if (num1 % 2 !=0 )
    { // If it is an odd number then add 1
        num1 = num1 + 1; }
    if(num2%2!=0)
    { num2=num2-1;

    }
    while (num1<=num2 )
    { // Write the condition part
        sum = sum + num1 ;
        num1=num1+2;
    }
    printf("The sum of even integers between the given limits = %d\n",sum  );
    return 0;
}
```



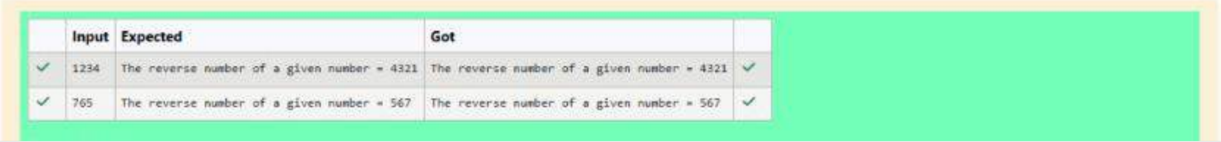
	Input	Expected	Got	
✓	3 6	The sum of even integers between the given limits = 10	The sum of even integers between the given limits = 10	✓

Passed all tests! ✓

Fill in the missing code in the below program to read an **integer number** and find the reverse of the given number.

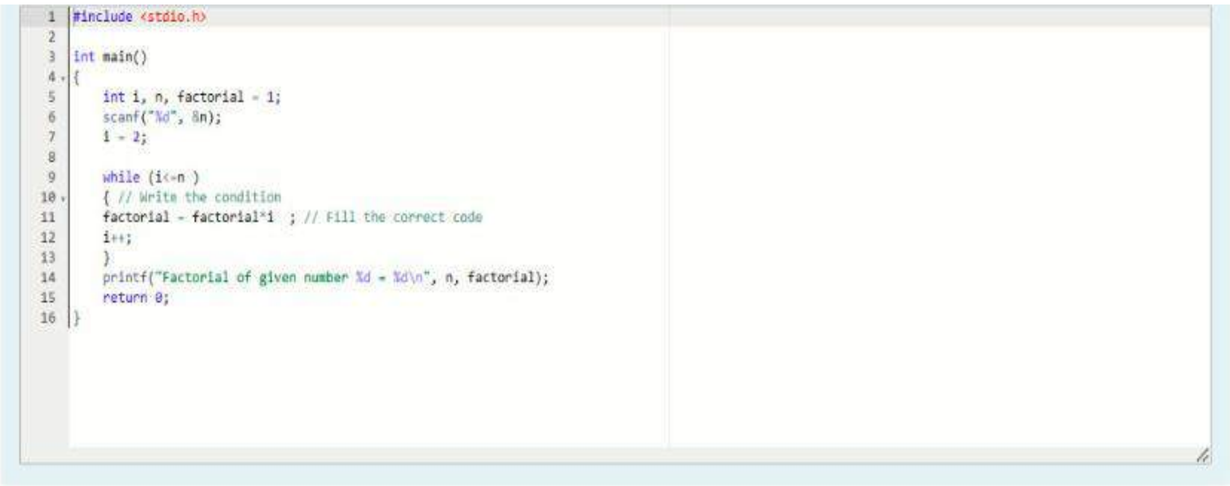


```
#include <stdio.h>

int main()
{
    int n, digit, reverse = 0;
    scanf("%d", &n);
    while ( n!=0 )
    { // Write the condition
    digit = n%10 ; // Fill the correct code
    reverse = reverse * 10 + digit ; // Fill the correct code
    n = n/10 ; // Fill the correct code
    }
    printf("The reverse number of a given number = %d" , reverse);
    return 0;
}
```

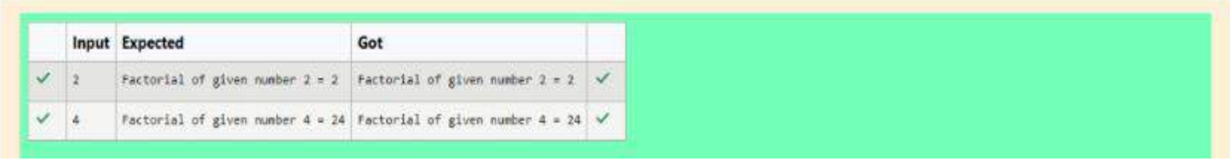


	Input	Expected	Got	
✓	1234	The reverse number of a given number = 4321	The reverse number of a given number = 4321	✓
✓	765	The reverse number of a given number = 567	The reverse number of a given number = 567	✓

Fill in the missing code in the below sample program which finds the factorial of a given number.

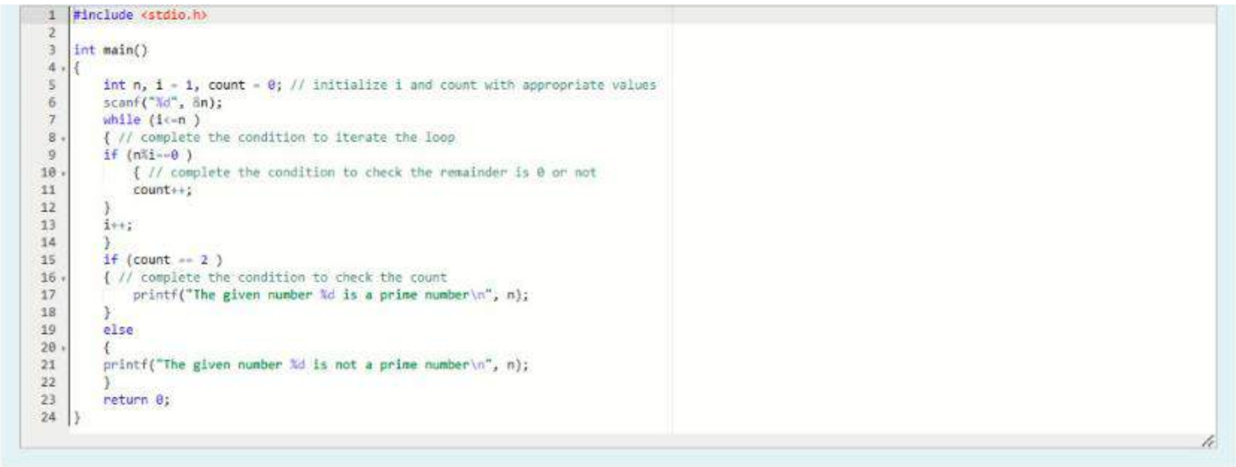


```c
#include <stdio.h>

int main()
{
    int i, n, factorial = 1;
    scanf("%d", &n);
    i = 2;

    while (i<=n )
    { // Write the condition
    factorial = factorial*i  ; // Fill the correct code
    i++;
    }
    printf("Factorial of given number %d = %d\n", n, factorial);
    return 0;
}
```

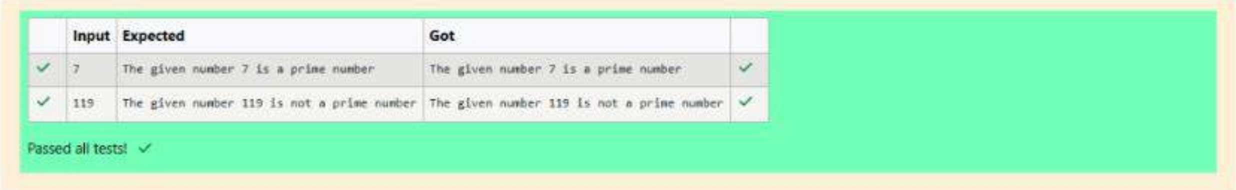


	Input	Expected	Got	
✓	2	Factorial of given number 2 = 2	Factorial of given number 2 = 2	✓
✓	4	Factorial of given number 4 = 24	Factorial of given number 4 = 24	✓

Below partial code is to verify if the given number is a prime number or not.



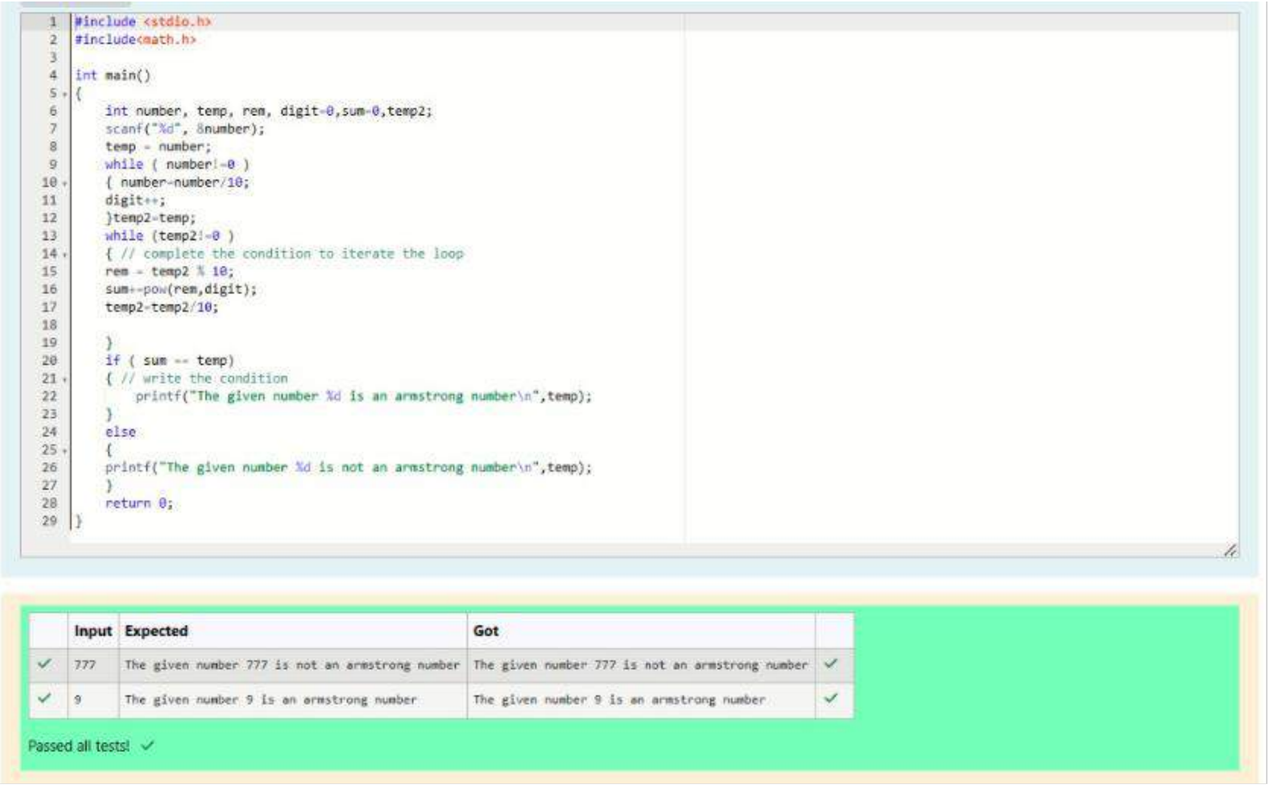
```c
#include <stdio.h>

int main()
{
    int n, i = 1, count = 0; // initialize i and count with appropriate values
    scanf("%d", &n);
    while (i<=n )
    { // complete the condition to iterate the loop
    if (n%i==0 )
        { // complete the condition to check the remainder is 0 or not
        count++;
    }
    i++;
    }
    if (count == 2 )
    { // complete the condition to check the count
        printf("The given number %d is a prime number\n", n);
    }
    else
    {
    printf("The given number %d is not a prime number\n", n);
    }
    return 0;
}
```



	Input	Expected	Got	
✓	7	The given number 7 is a prime number	The given number 7 is a prime number	✓
✓	119	The given number 119 is not a prime number	The given number 119 is not a prime number	✓

Passed all tests! ✓

Below partial code is to verify if the given number is an armstrong number or not.

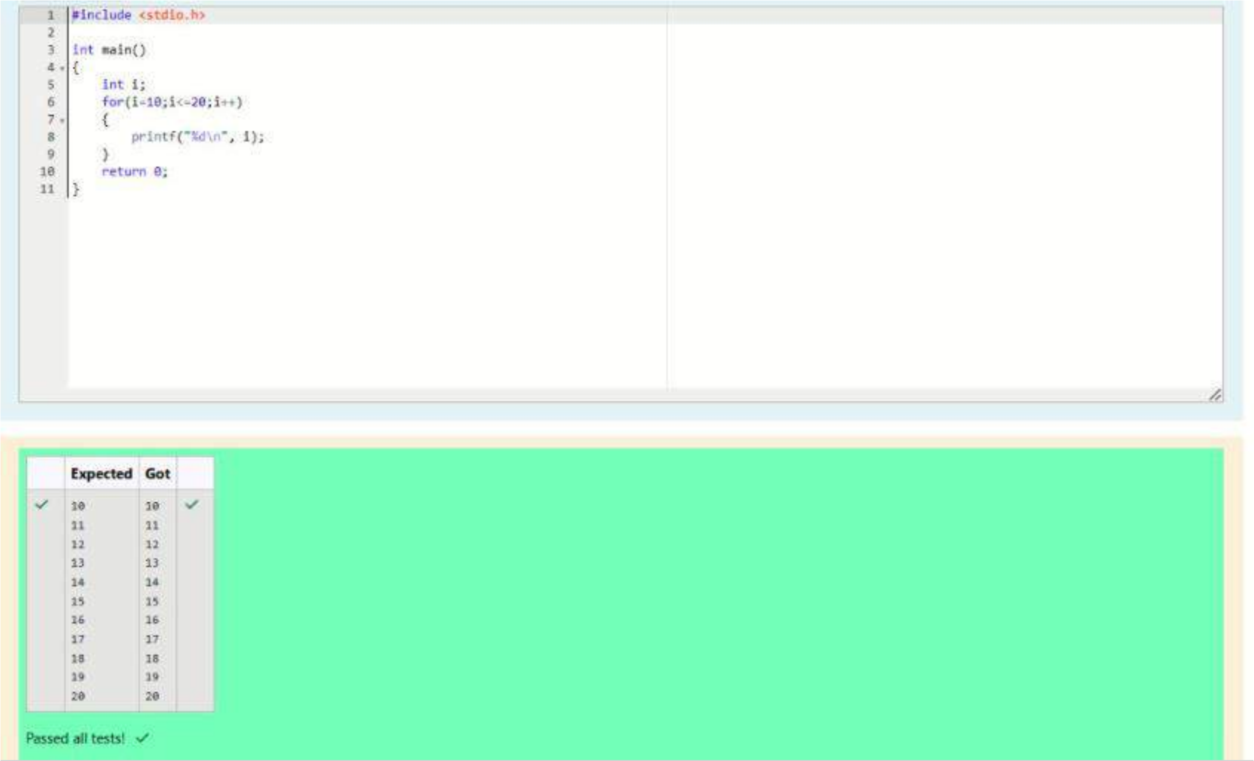


```c
#include <stdio.h>
#include<math.h>

int main()
{
    int number, temp, rem, digit=0,sum=0,temp2;
    scanf("%d", &number);
    temp = number;
    while ( number!=0 )
    { number=number/10;
    digit++;
    }temp2=temp;
    while (temp2!=0 )
    { // complete the condition to iterate the loop
    rem = temp2 % 10;
    sum+=pow(rem,digit);
    temp2=temp2/10;

    }
    if ( sum == temp)
    { // write the condition
        printf("The given number %d is an armstrong number\n",temp);
    }
    else
    {
    printf("The given number %d is not an armstrong number\n",temp);
    }
    return 0;
}
```

	Input	Expected	Got	
✓	777	The given number 777 is not an armstrong number	The given number 777 is not an armstrong number	✓
✓	9	The given number 9 is an armstrong number	The given number 9 is an armstrong number	✓

Passed all tests! ✓

For-loop

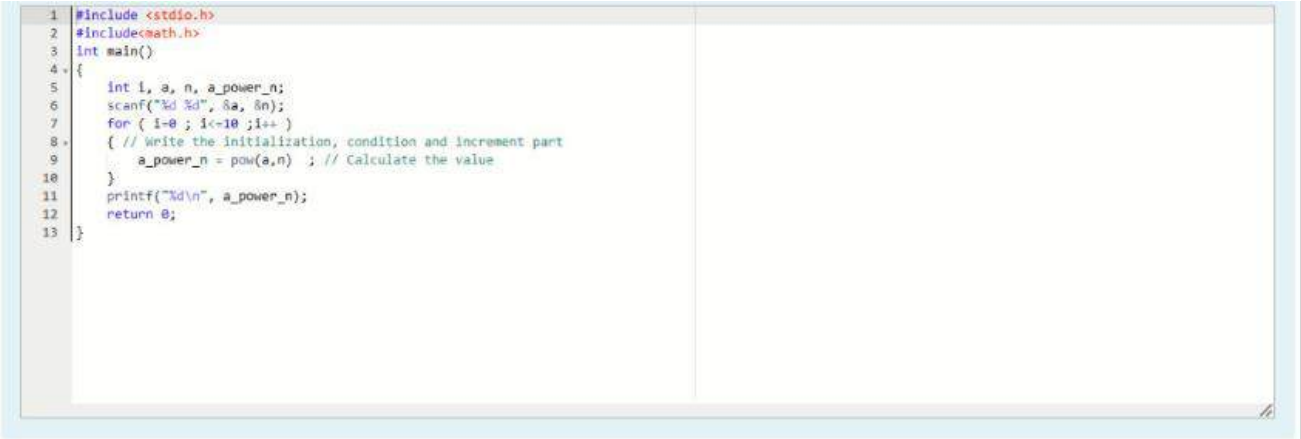


```c
#include <stdio.h>

int main()
{
    int i;
    for(i=10;i<=20;i++)
    {
        printf("%d\n", i);
    }
    return 0;
}
```

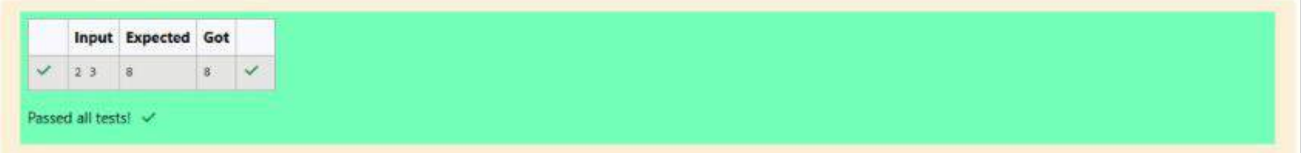
	Expected	Got	
✓	10 11 12 13 14 15 16 17 18 19 20	10 11 12 13 14 15 16 17 18 19 20	✓

Passed all tests! ✓

Fill in the missing code in the below program to calculate the value of a^n, given two positive non-zero integers a and n.



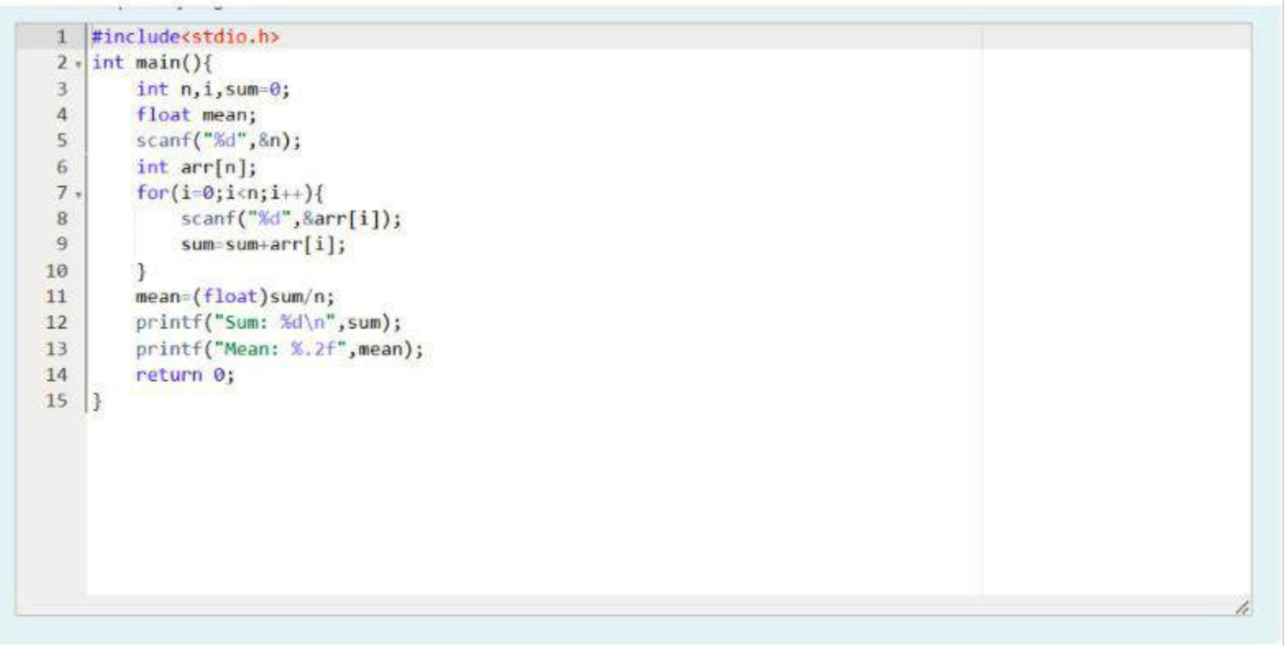
```c
#include <stdio.h>
#include<math.h>
int main()
{
    int i, a, n, a_power_n;
    scanf("%d %d", &a, &n);
    for ( i=0 ; i<=10 ;i++ )
    { // Write the initialization, condition and increment part
        a_power_n = pow(a,n)  ; // Calculate the value
    }
    printf("%d\n", a_power_n);
    return 0;
}
```



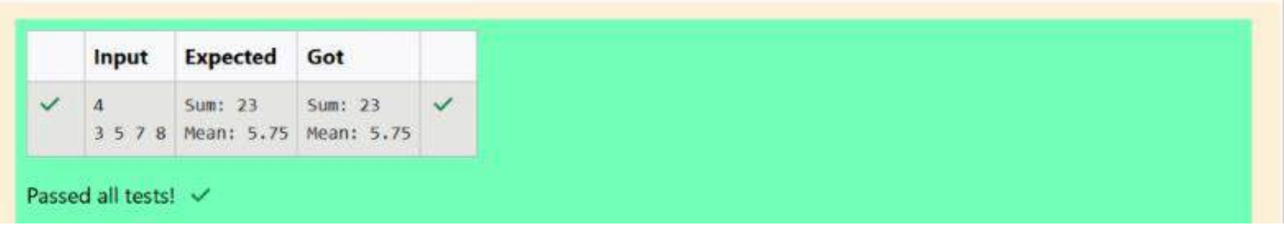
	Input	Expected	Got	
✓	2 3	8	8	✓

Passed all tests! ✓

Write a program to find **sum** and **mean** of **n** numbers.



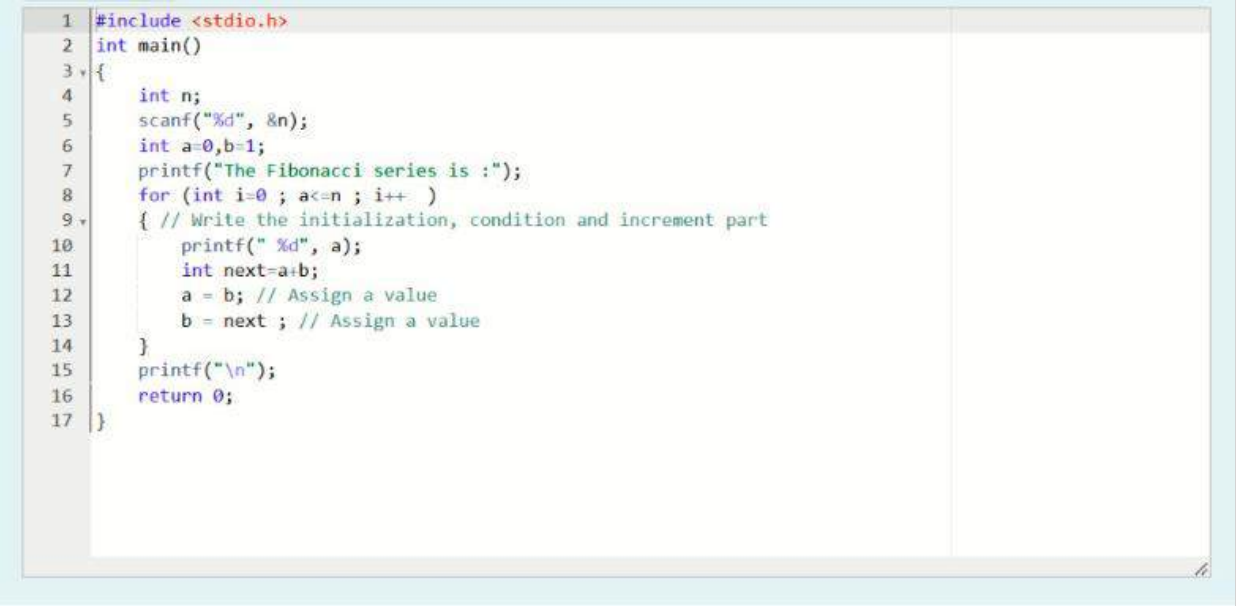
```c
#include<stdio.h>
int main(){
    int n,i,sum=0;
    float mean;
    scanf("%d",&n);
    int arr[n];
    for(i=0;i<n;i++){
        scanf("%d",&arr[i]);
        sum=sum+arr[i];
    }
    mean=(float)sum/n;
    printf("Sum: %d\n",sum);
    printf("Mean: %.2f",mean);
    return 0;
}
```



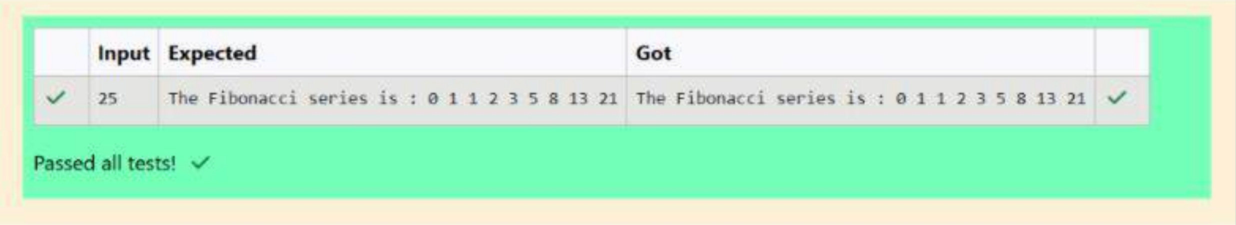
	Input	Expected	Got	
✓	4 3 5 7 8	Sum: 23 Mean: 5.75	Sum: 23 Mean: 5.75	✓

Passed all tests! ✓

Fill in the missing code in the below program to print the Fibonacci series i.e., 0 1 1 2 3 5 8 13 21....., up to the limit.



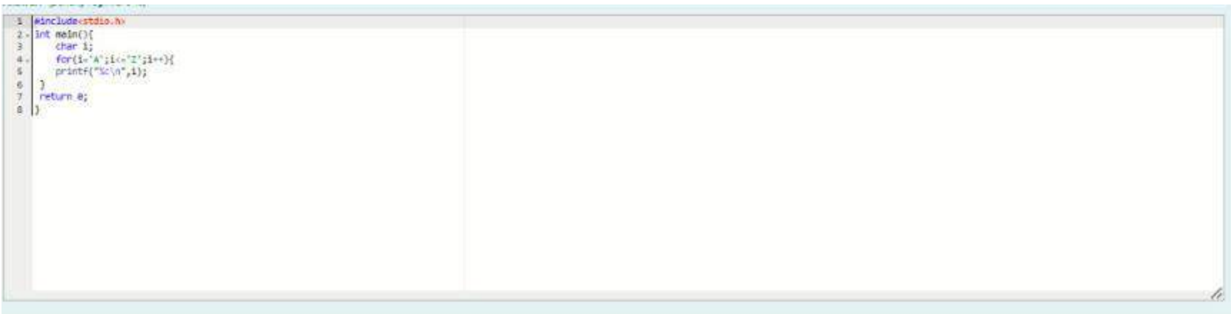
```c
#include <stdio.h>
int main()
{
    int n;
    scanf("%d", &n);
    int a=0,b=1;
    printf("The Fibonacci series is :");
    for (int i=0 ; a<=n ; i++  )
    { // Write the initialization, condition and increment part
        printf(" %d", a);
        int next=a+b;
        a = b; // Assign a value
        b = next ; // Assign a value
    }
    printf("\n");
    return 0;
}
```



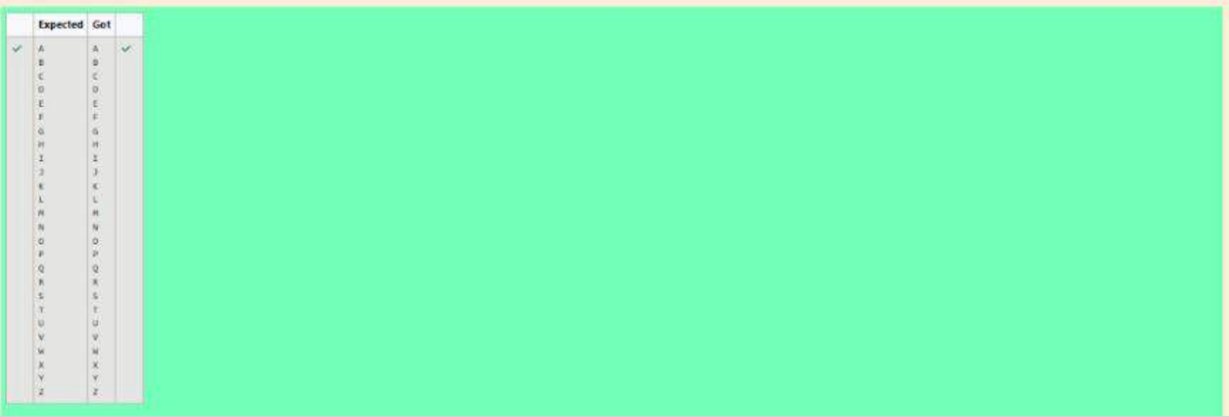
	Input	Expected	Got	
✓	25	The Fibonacci series is : 0 1 1 2 3 5 8 13 21	The Fibonacci series is : 0 1 1 2 3 5 8 13 21	✓

Passed all tests! ✓

Write a program that will print all the **English alphabets** from A to Z, each in a new line.

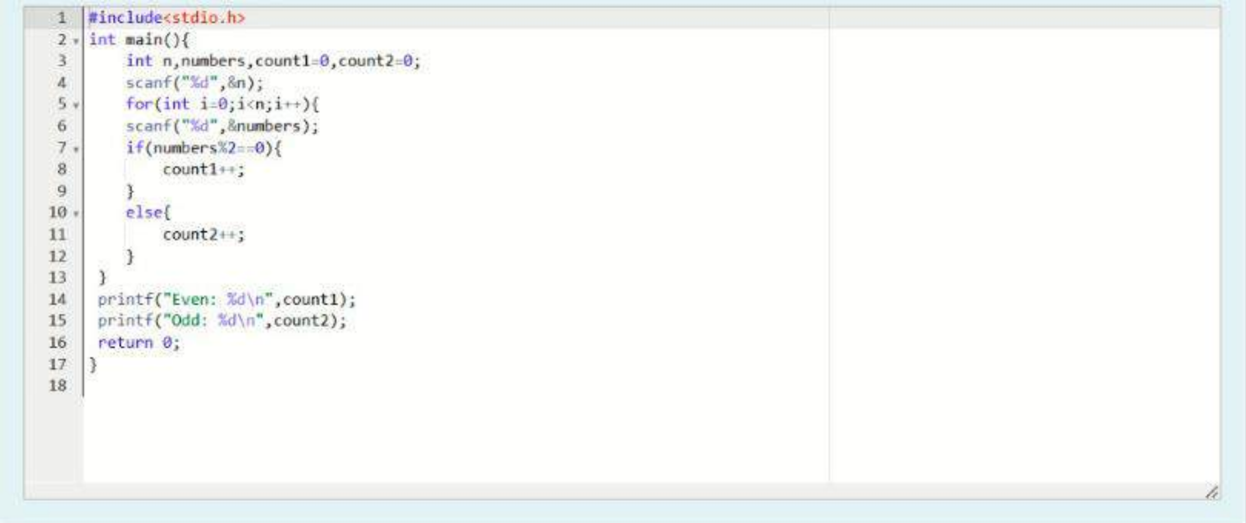


```c
#include<stdio.h>
int main(){
    char i;
    for(i='A';i<='Z';i++){
    printf("%c\n",i);
 }
 return 0;
}
```

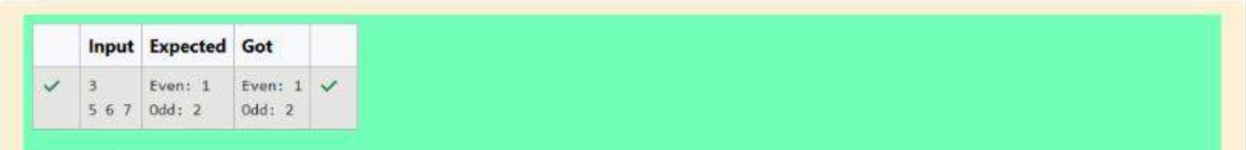


	Expected	Got	
✓	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z	✓

Write a program to read **n** numbers from the user and then count number of "**Odd"** and "**Even"** numbers.

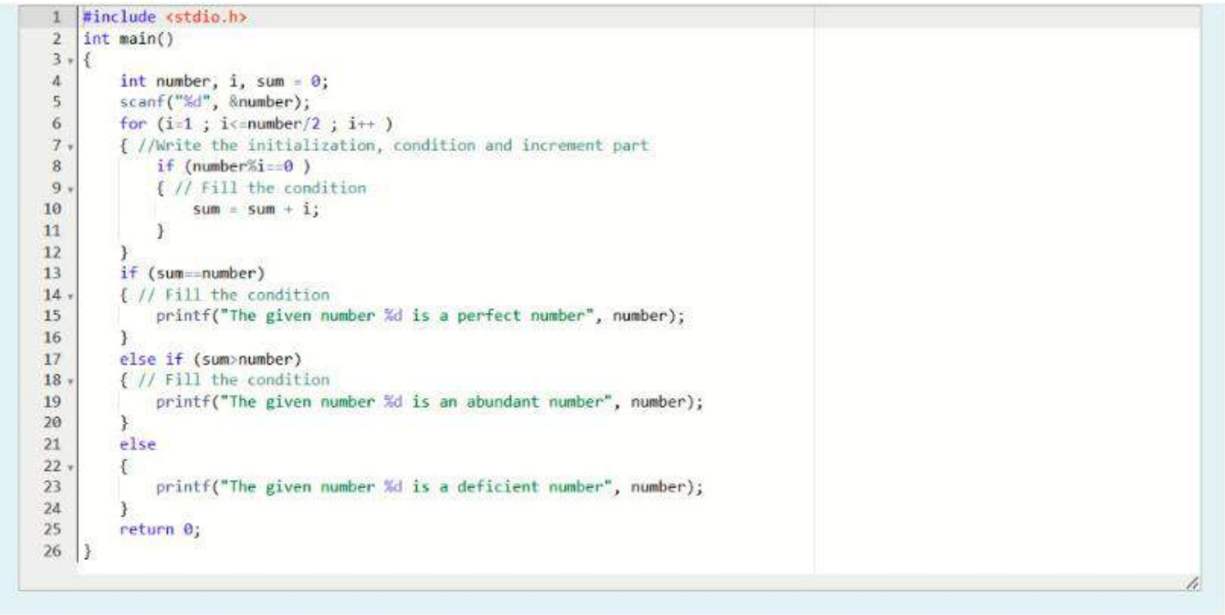


```c
#include<stdio.h>
int main(){
    int n,numbers,count1=0,count2=0;
    scanf("%d",&n);
    for(int i=0;i<n;i++){
    scanf("%d",&numbers);
    if(numbers%2==0){
        count1++;
    }
    else{
        count2++;
    }
 }
 printf("Even: %d\n",count1);
 printf("Odd: %d\n",count2);
 return 0;
}
```

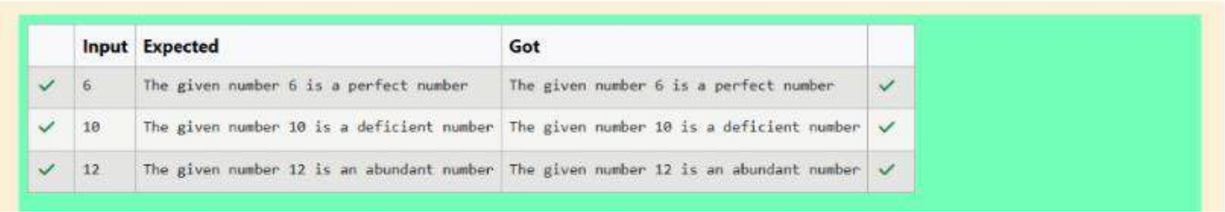


	Input	Expected	Got	
✓	3 5 6 7	Even: 1 Odd: 2	Even: 1 Odd: 2	✓

Fill in the missing code in the below program to verify whether the given number is perfect, abundant or deficient.

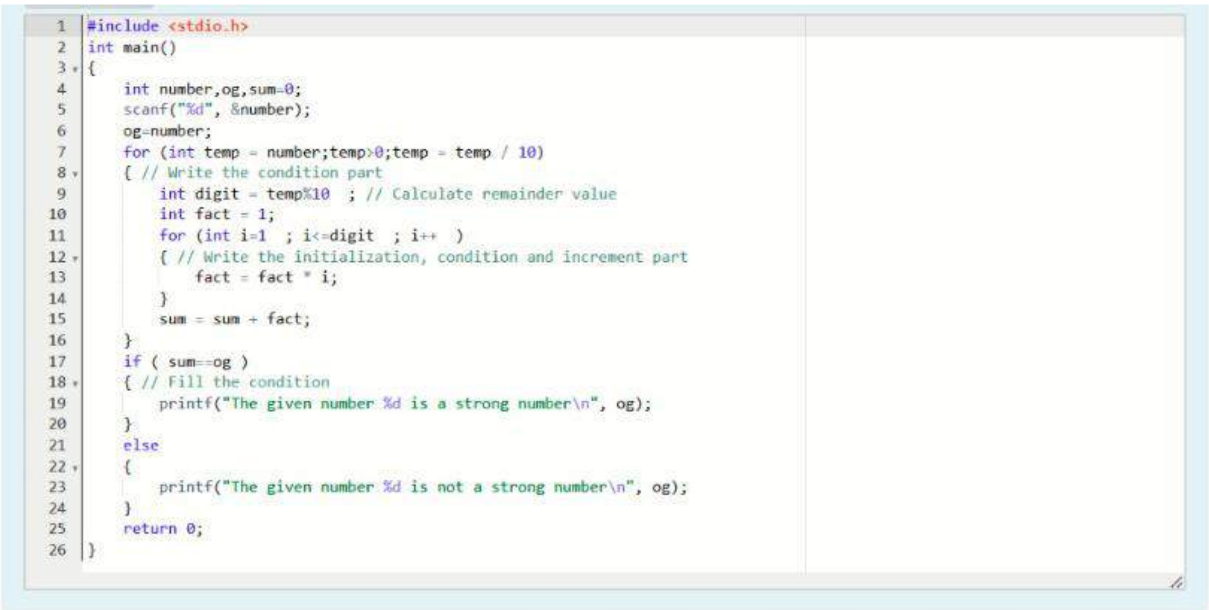


```c
#include <stdio.h>
int main()
{
    int number, i, sum = 0;
    scanf("%d", &number);
    for (i=1 ; i<=number/2 ; i++ )
    { //Write the initialization, condition and increment part
        if (number%i==0 )
        { // Fill the condition
            sum = sum + i;
        }
    }
    if (sum==number)
    { // Fill the condition
        printf("The given number %d is a perfect number", number);
    }
    else if (sum>number)
    { // Fill the condition
        printf("The given number %d is an abundant number", number);
    }
    else
    {
        printf("The given number %d is a deficient number", number);
    }
    return 0;
}
```

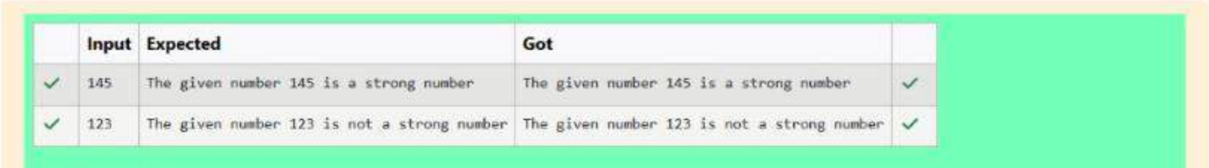


	Input	Expected	Got	
✓	6	The given number 6 is a perfect number	The given number 6 is a perfect number	✓
✓	10	The given number 10 is a deficient number	The given number 10 is a deficient number	✓
✓	12	The given number 12 is an abundant number	The given number 12 is an abundant number	✓

Fill in the missing code in the below program to check whether the given number is a strong number or not.



```c
#include <stdio.h>
int main()
{
    int number,og,sum=0;
    scanf("%d", &number);
    og=number;
    for (int temp = number;temp>0;temp = temp / 10)
    { // Write the condition part
        int digit = temp%10  ; // Calculate remainder value
        int fact = 1;
        for (int i=1  ; i<=digit  ; i++  )
        { // Write the initialization, condition and increment part
            fact = fact * i;
        }
        sum = sum + fact;
    }
    if ( sum==og )
    { // Fill the condition
        printf("The given number %d is a strong number\n", og);
    }
    else
    {
        printf("The given number %d is not a strong number\n", og);
    }
    return 0;
}
```



	Input	Expected	Got	
✓	145	The given number 145 is a strong number	The given number 145 is a strong number	✓
✓	123	The given number 123 is not a strong number	The given number 123 is not a strong number	✓