NAME : SREYA G **REGISTER NO.:** 230701334

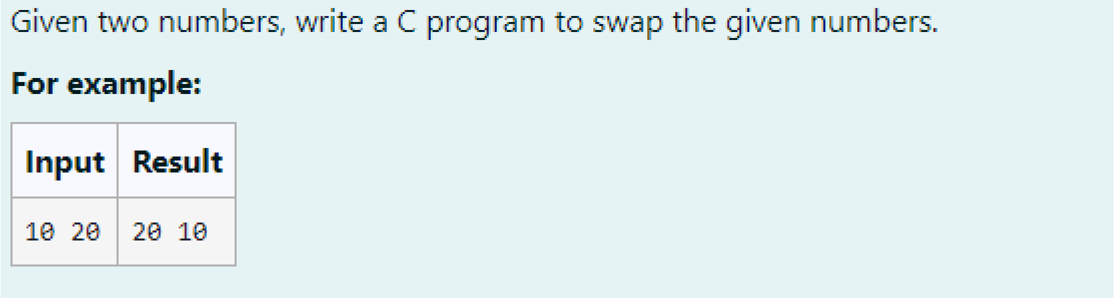
CLASS : CSE F **DATE:** 01/08/2024

EX-1:

BASIC C PROGRAMMING:

PROBLEM 1:

AIM:



Given two numbers, write a C program to swap the given numbers.

For example:

Input	Result
10 20	20 10

ALGORITHM:

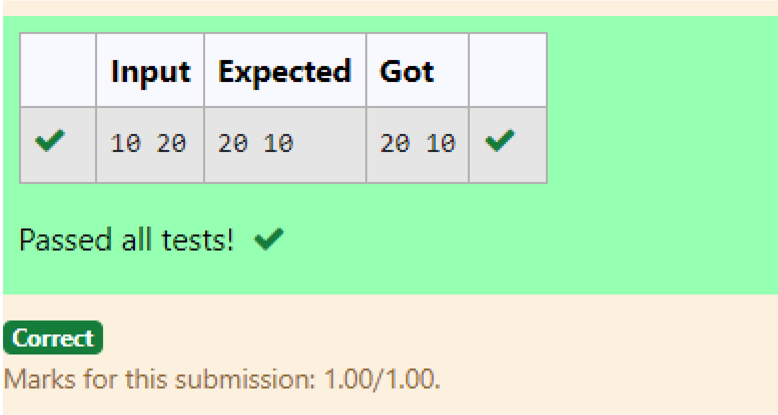
1. Input two integers a and b.
2. Store a in a temporary variable temp.
3. Assign the value of b to a.
4. Assign the value of temp (original a) to b.
5. Output the swapped values of a and b.

CODE:

```
#include<stdio.h>
int main()
{
   int a,b,temp;
   scanf("%d",&a);
   scanf("%d",&b);
   temp =a;
```

```
    a = b;
    b = temp;
    printf("%d ",a);
    printf("%d",b);
}
```

OUTPUT:



	Input	Expected	Got	
✓	10 20	20 10	20 10	✓

Passed all tests! ✓

Correct

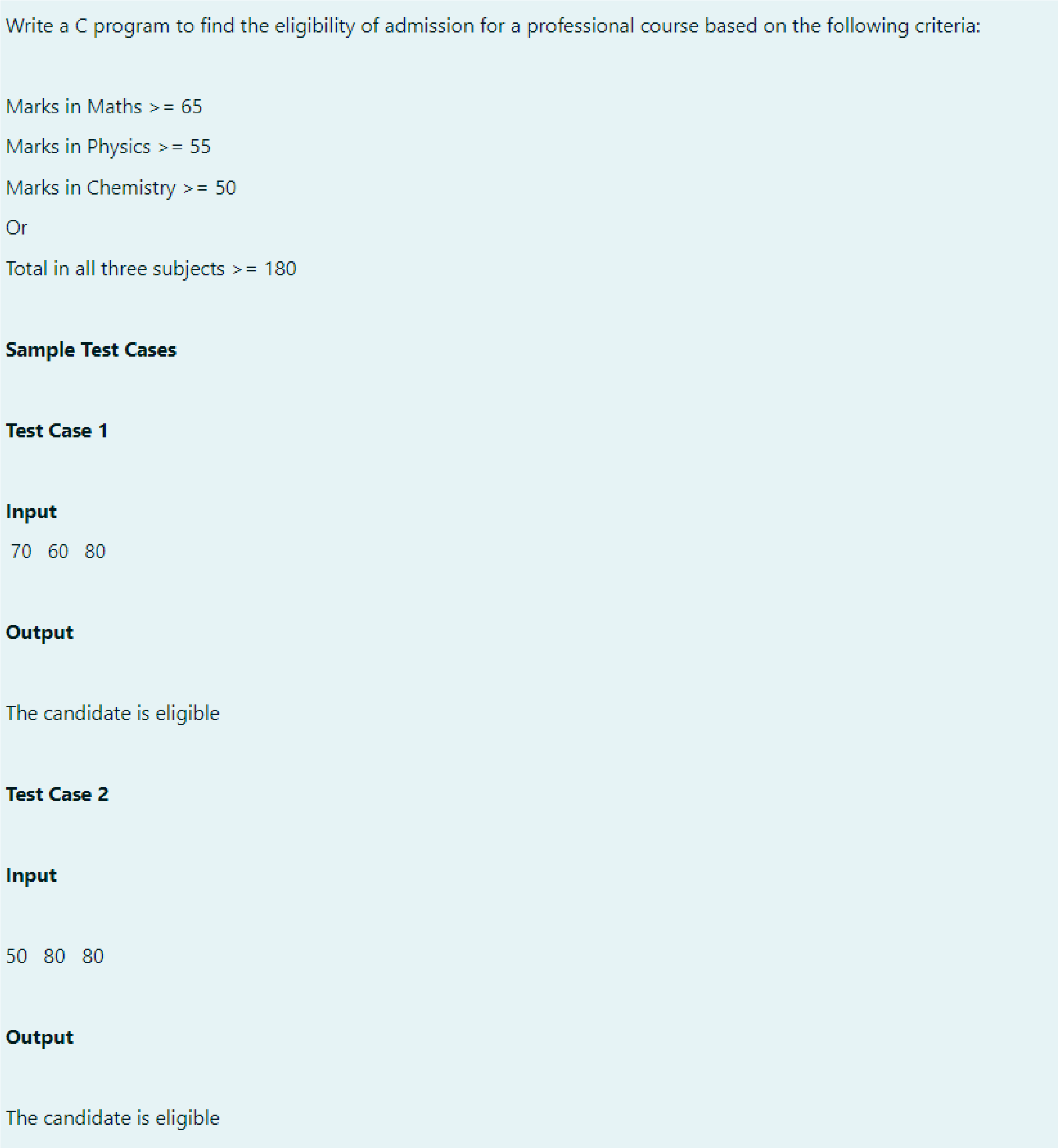
Marks for this submission: 1.00/1.00.

RESULT:

Thus the code is executed successfully and gives the expected output.

PROBLEM 2:

AIM:



Write a C program to find the eligibility of admission for a professional course based on the following criteria:

Marks in Maths >= 65

Marks in Physics >= 55

Marks in Chemistry >= 50

Or

Total in all three subjects >= 180

Sample Test Cases

Test Case 1

Input

```
 70   60   80
```

Output

```
The candidate is eligible
```

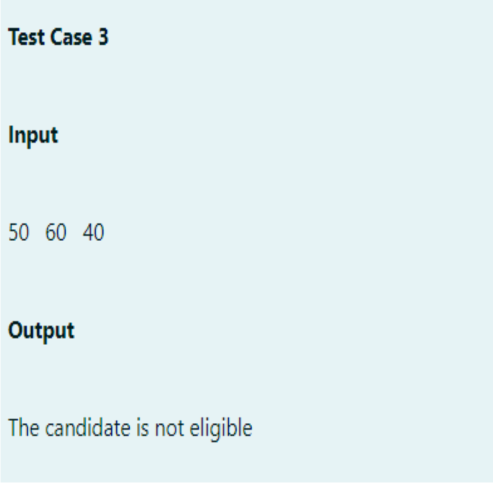
Test Case 2

Input

```
50   80   80
```

Output

```
The candidate is eligible
```



Test Case 3

Input

```
50   60   40
```

Output

```
The candidate is not eligible
```

ALGORITHM:

1. Input the marks for maths,physics, and chemistry.

2. Calculate the total marks by adding maths, physics, and chemistry.

3. Check eligibility:

   - If maths >= 65, physics >= 55, and chemistry >= 50, the candidate is eligible.

   - Else if the total marks are greater than or equal to 180, the candidate is eligible.

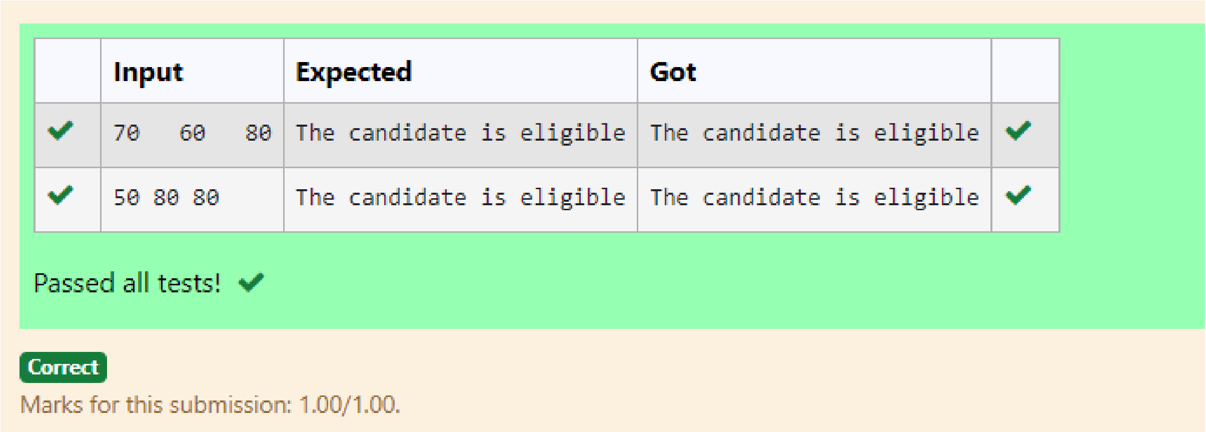
4. Output eligibility status: Print "The candidate is eligible" or "The candidate is not eligible".

CODE:

```
#include<stdio.h>
int main()
{
   int maths,physics,chemistry;
   int total;
   scanf("%d",&maths);
   scanf("%d",&physics);
   scanf("%d",&chemistry);
   total = maths + physics + chemistry;
   if(maths >= 65 && physics >= 55 && chemistry >= 50)
   {
      printf("The candidate is eligible");
   }
   else if(total >= 180)
   {
      printf("The candidate is eligible");
   }
   else
   {
      printf("The candidate is not eligible");
   }
```

}

OUTPUT:



	Input	Expected	Got	
✓	70 60 80	The candidate is eligible	The candidate is eligible	✓
✓	50 80 80	The candidate is eligible	The candidate is eligible	✓

Passed all tests! ✓

Correct

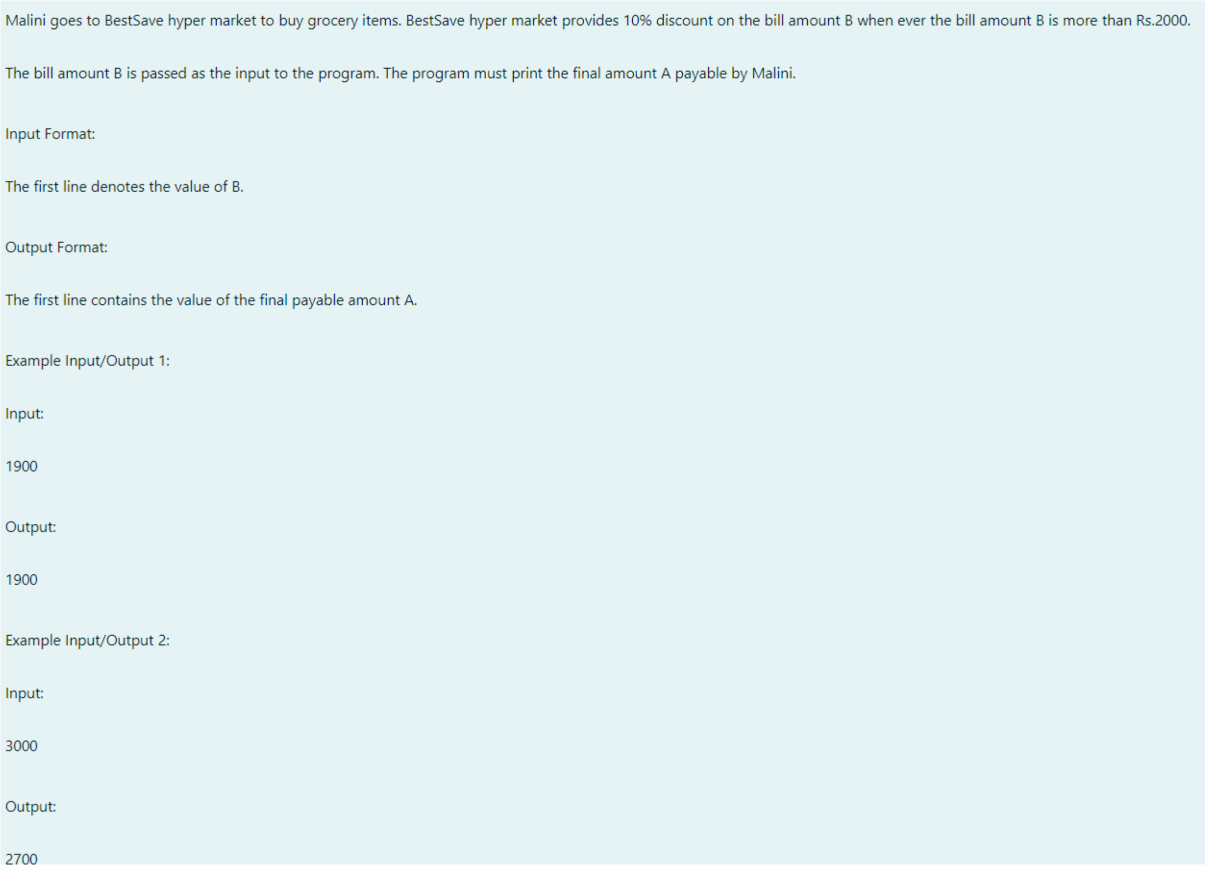
Marks for this submission: 1.00/1.00.

RESULT:

Thus the code is executed successfully and gives the expected output.

PROBLEM 3:

AIM:



Malini goes to BestSave hyper market to buy grocery items. BestSave hyper market provides 10% discount on the bill amount B when ever the bill amount B is more than Rs.2000.

The bill amount B is passed as the input to the program. The program must print the final amount A payable by Malini.

Input Format:

The first line denotes the value of B.

Output Format:

The first line contains the value of the final payable amount A.

Example Input/Output 1:

Input:

1900

Output:

1900

Example Input/Output 2:

Input:

3000

Output:

2700

ALGORITHM:

```
1. Input the value B.
2. If B > 2000:
   - Calculate c = 0.1 * B (10% of B).
   - Calculate d = B - c (remaining amount after deduction).
   - Output d.
3. Else, output B.
```

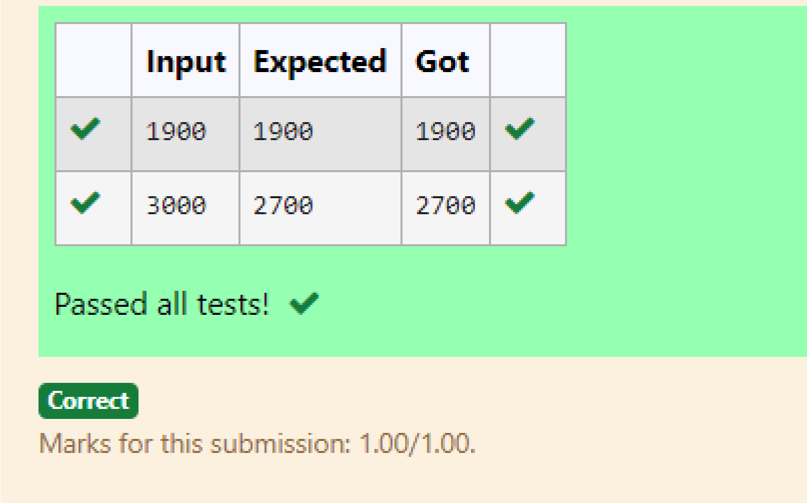
CODE:

```
#include<stdio.h>
int main()
{
    int B;
    scanf("%d",&B);
    if(B > 2000)
```

```
  {
     int c,d;
     c = ((0.1)*B);
     d= B-c;
     printf("%d",d);
  }
  else
  {
     printf("%d",B);
  }

}
```

OUTPUT:



	Input	Expected	Got	
✓	1900	1900	1900	✓
✓	3000	2700	2700	✓

Passed all tests! ✓

Correct

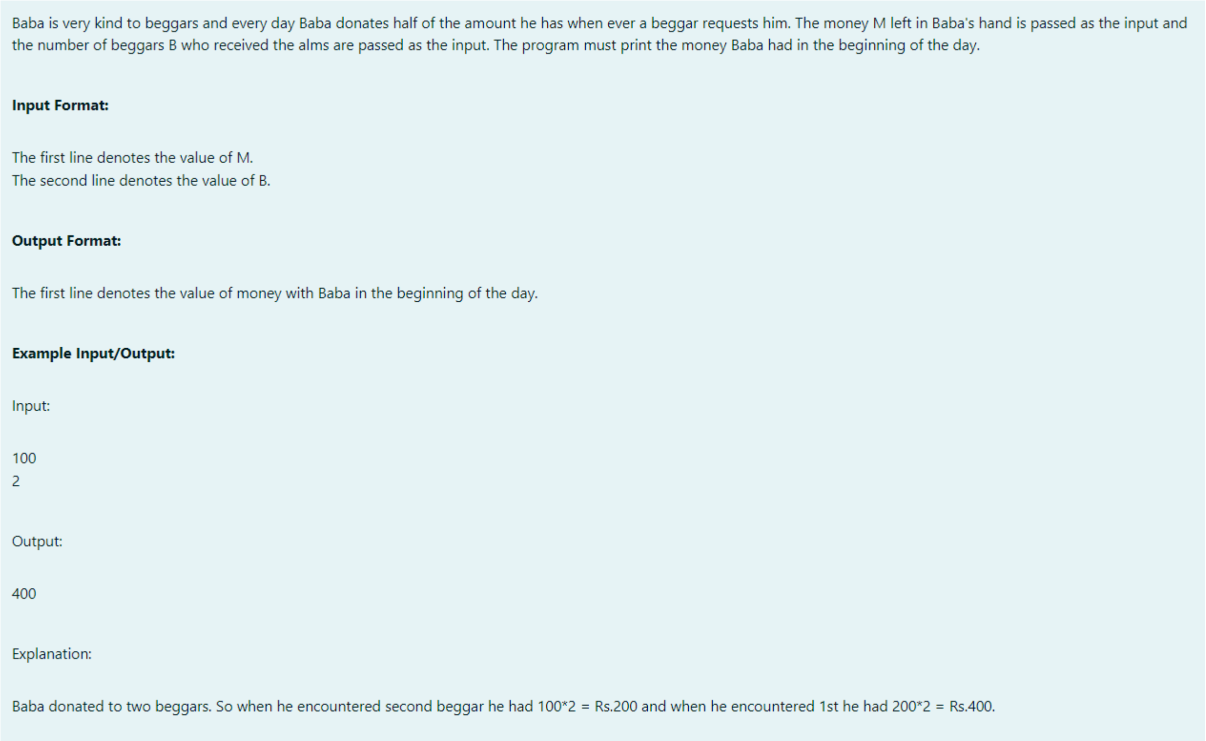
Marks for this submission: 1.00/1.00.

RESULT:

Thus the code is executed successfully and gives the expected output.

PROBLEM 4:

AIM:



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.

Input Format:

The first line denotes the value of M.
The second line denotes the value of B.

Output Format:

The first line denotes the value of money with Baba in the beginning of the day.

Example Input/Output:

Input:

100
2

Output:

400

Explanation:

Baba donated to two beggars. So when he encountered second beggar he had 100*2 = Rs.200 and when he encountered 1st he had 200*2 = Rs.400.

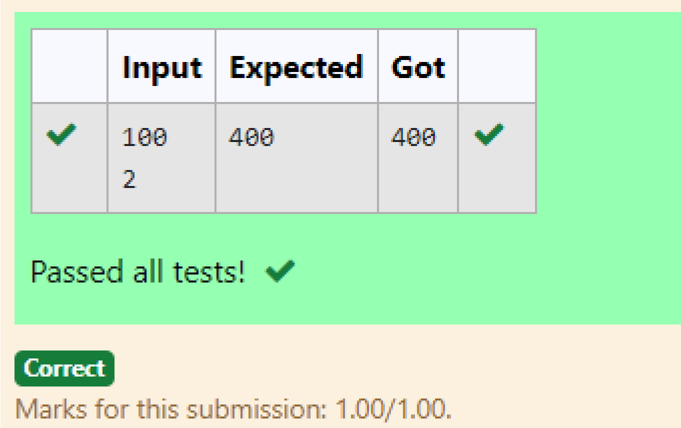
ALGORITHM:

```
1. Input M and B.
2. Repeat B times:
   - Multiply M by 2.
3. Output M.
```

CODE:

```
#include<stdio.h>
int main()
{
    int M,B;
    scanf("%d",&M);
    scanf("%d",&B);
    int i = 0;
    while(i < B)
    {
        M = M * 2;
        i++;
    }
    printf("%d",M);
}
```

OUTPUT:



	Input	Expected	Got	
✓	100 2	400	400	✓

Passed all tests! ✓

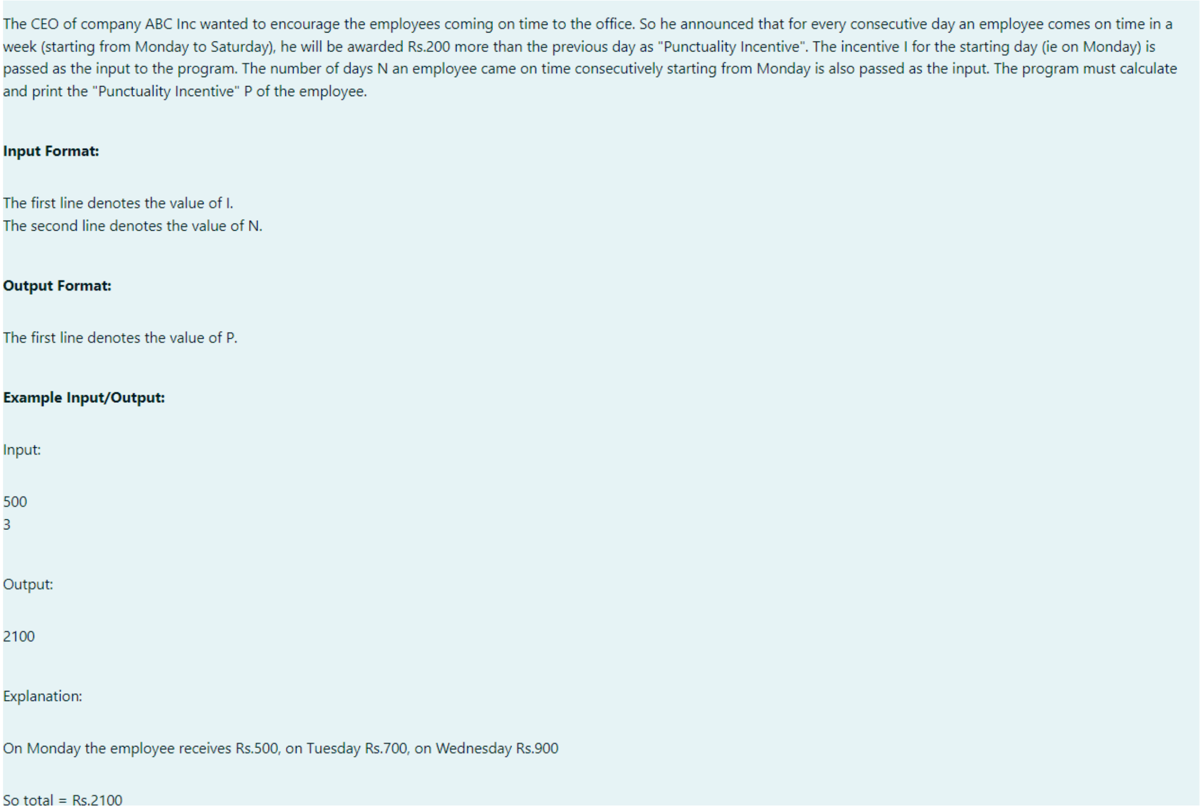
Marks for this submission: 1.00/1.00.

RESULT:

Thus the code is executed successfully and gives the expected output.

PROBLEM 5:

AIM:



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.

Input Format:

The first line denotes the value of I.
The second line denotes the value of N.

Output Format:

The first line denotes the value of P.

Example Input/Output:

Input:

```
500
3
```

Output:

```
2100
```

Explanation:

On Monday the employee receives Rs.500, on Tuesday Rs.700, on Wednesday Rs.900

So total = Rs.2100

ALGORITHM:

1. Input I and N.

2. Initialize t = 0 and a = 0.

3. Repeat N times:

   - Add I to t.

   - Increase I by 200.

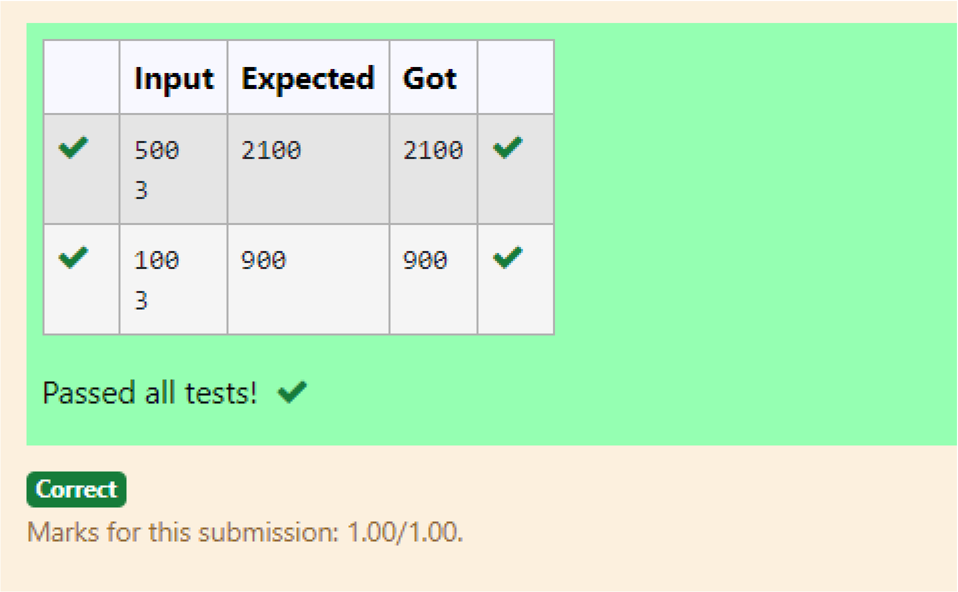
4. **Output t.

CODE:

```
#include<stdio.h>
int main()
{
    int I,N;
    int t = 0 , a = 0;
    scanf("%d",&I);
    scanf("%d",&N);
    while(a < N)
    {
        t = t + I;
        I = I + 200;
        a++;
    }
    printf("%d",t);

}
```

OUTPUT:



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

Passed all tests! ✓

Correct

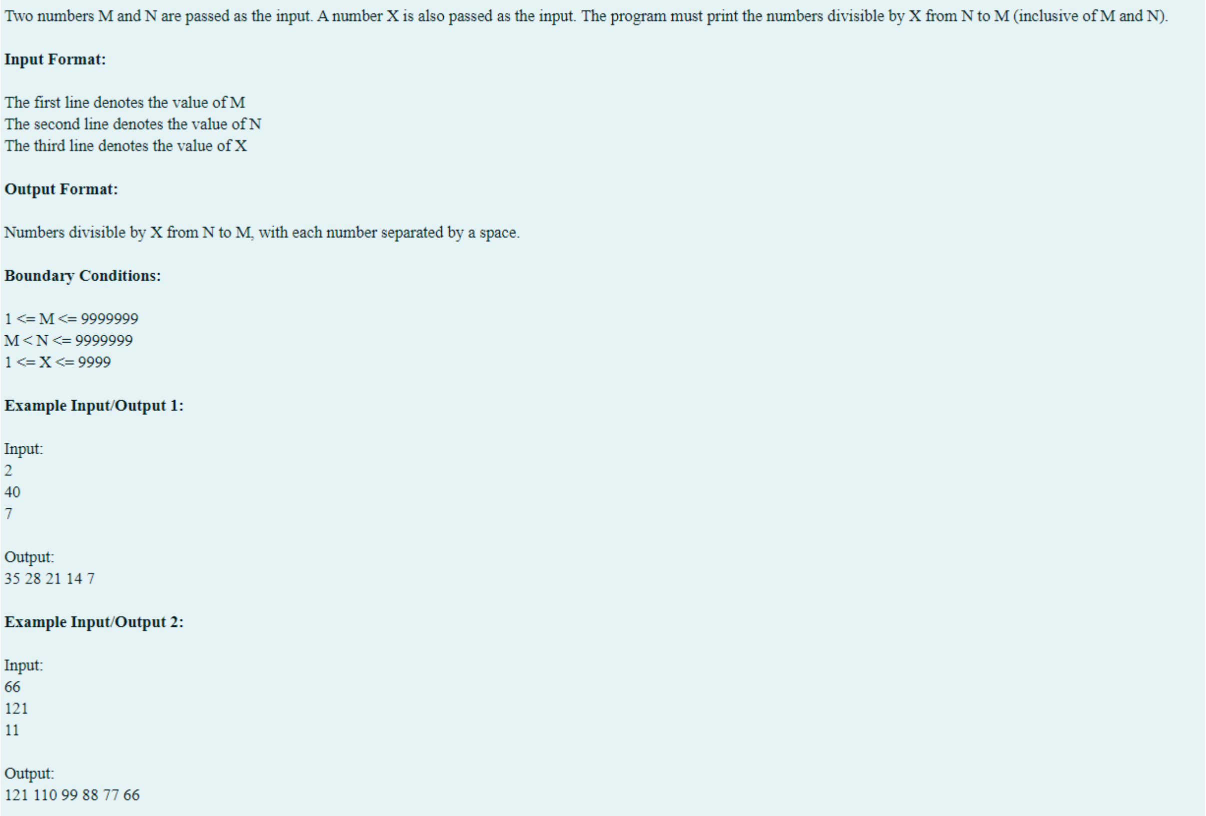
Marks for this submission: 1.00/1.00.

RESULT:

Thus the code is executed successfully and gives the expected output.

PROBLEM 6:

AIM:



Two numbers M and N are passed as the input. A number X is also passed as the input. The program must print the numbers divisible by X from N to M (inclusive of M and N).

Input Format:

The first line denotes the value of M
The second line denotes the value of N
The third line denotes the value of X

Output Format:

Numbers divisible by X from N to M, with each number separated by a space.

Boundary Conditions:

1 <= M <= 9999999
M < N <= 9999999
1 <= X <= 9999

Example Input/Output 1:

```
Input:
2
40
7

Output:
35 28 21 14 7
```

Example Input/Output 2:

```
Input:
66
121
11

Output:
121 110 99 88 77 66
```

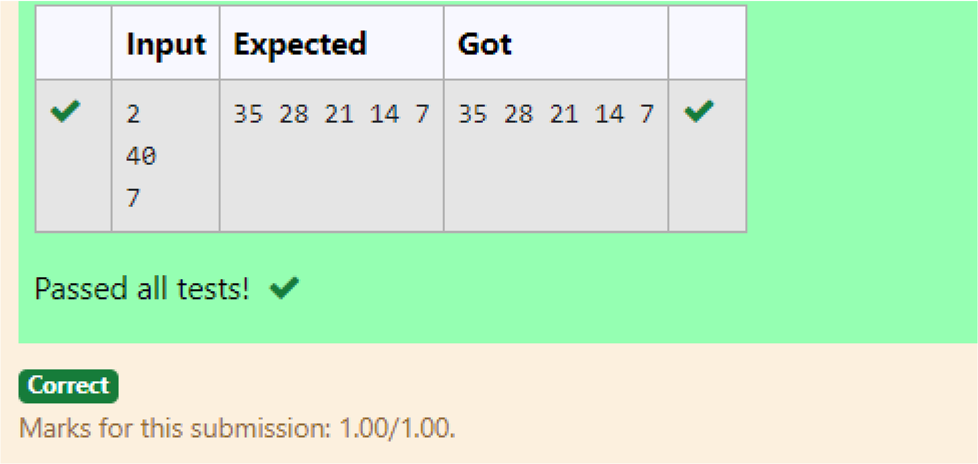
ALGORITHM:

```
1. Input a, b, and c.
2. Loop from b to a:
   - If i % c == 0, print i.
3. End.
```

CODE:

```
#include<stdio.h>
int main()
{
    int a,b,c;
    scanf("%d %d %d",&a,&b,&c);
    for(int i = b;i >= a;i--)
    {
        if(i % c == 0)
        {
            printf("%d ",i);
        }
    }
}
```

OUTPUT:



	Input	Expected	Got	
✓	2 40 7	35 28 21 14 7	35 28 21 14 7	✓

Passed all tests! ✓

Correct

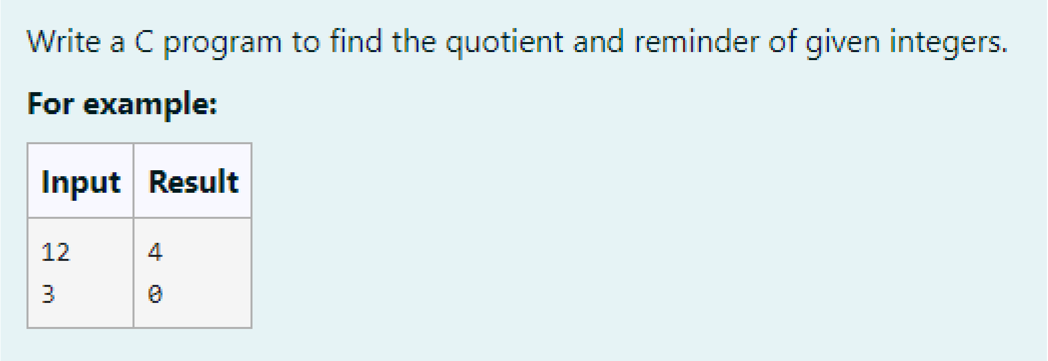
Marks for this submission: 1.00/1.00.

RESULT:

Thus the code is executed successfully and gives the expected output.

PROBLEM 7:

AIM:



Write a C program to find the quotient and reminder of given integers.

For example:

Input	Result
12 3	4 0

ALGORITHM:

1. Input num1 and num2.

2. Calculate the quotient: q = num1 / num2.

3. Print q.

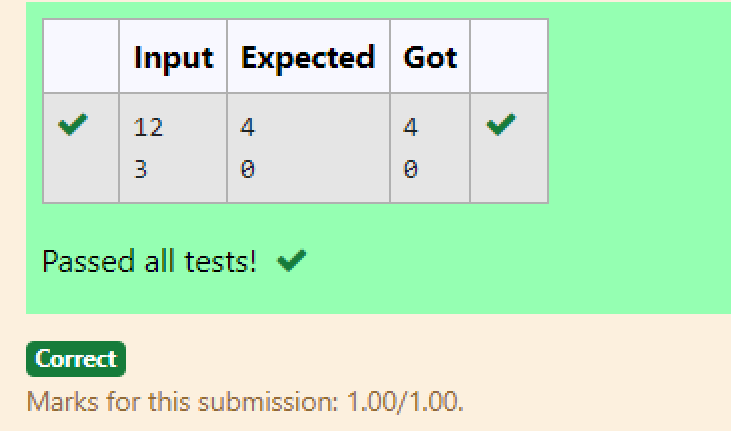
4. Calculate the remainder: r = num1 % num2.

5. Print r.

CODE:

```
#include<stdio.h>
int main()
{
   int num1;
   int num2;
   scanf("%d",&num1);
   scanf("%d",&num2);
   int q = num1 / num2;
   printf("%d \n",q);
   int r = num1 % num2;
   printf("%d",r);
}
```

OUTPUT:



	Input	Expected	Got	
✓	12 3	4 0	4 0	✓

Passed all tests! ✓

Correct

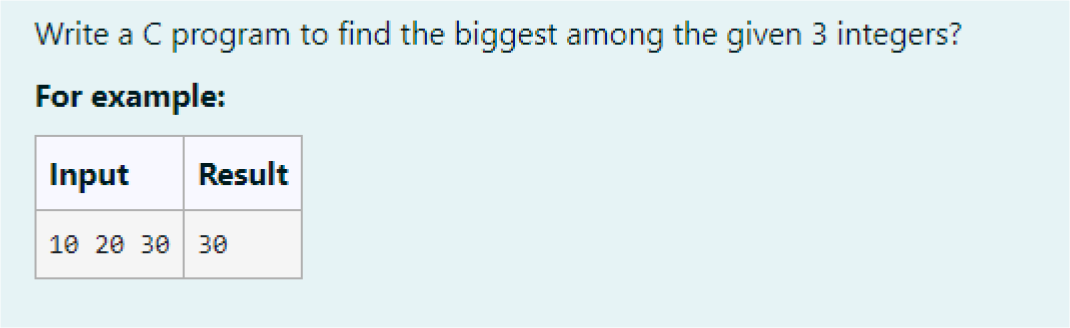
Marks for this submission: 1.00/1.00.

RESULT:

Thus the code is executed successfully and gives the expected output.

PROBLEM 8:

AIM:



Write a C program to find the biggest among the given 3 integers?

For example:

Input	Result
`10 20 30`	`30`

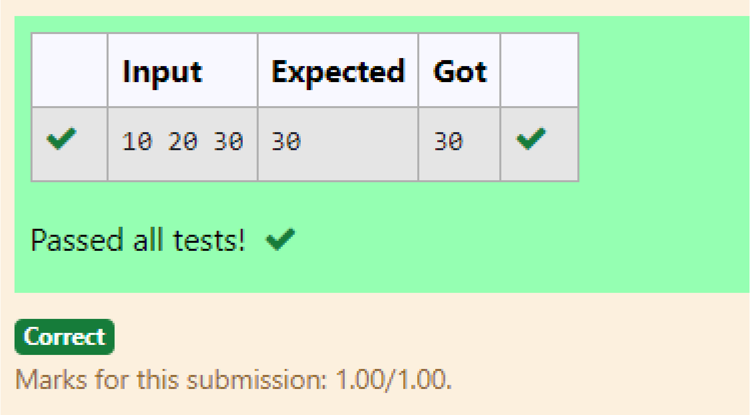
ALGORITHM:

```
1. Input num1, num2, and num3.
2. Compare the numbers:
  - If num1 is greater than both num2 and num3, print num1.
  - Else if num2 is greater than both num1 and num3, print num2.
  - Else, print num3.
```

CODE:

```
#include<stdio.h>
int main()
{
   int num1,num2,num3;
   scanf("%d %d %d",&num1,&num2,&num3);
   if(num1 > num2 && num1 > num3)
   {
      printf("%d",num1);
   }
   else if(num2 > num1 && num2 >num3)
   {
      printf("%d",num2);
   }
   else
   {
        printf("%d",num3);
    }
}
```

OUTPUT:



	Input	Expected	Got	
✓	10 20 30	30	30	✓

Passed all tests! ✓

Correct

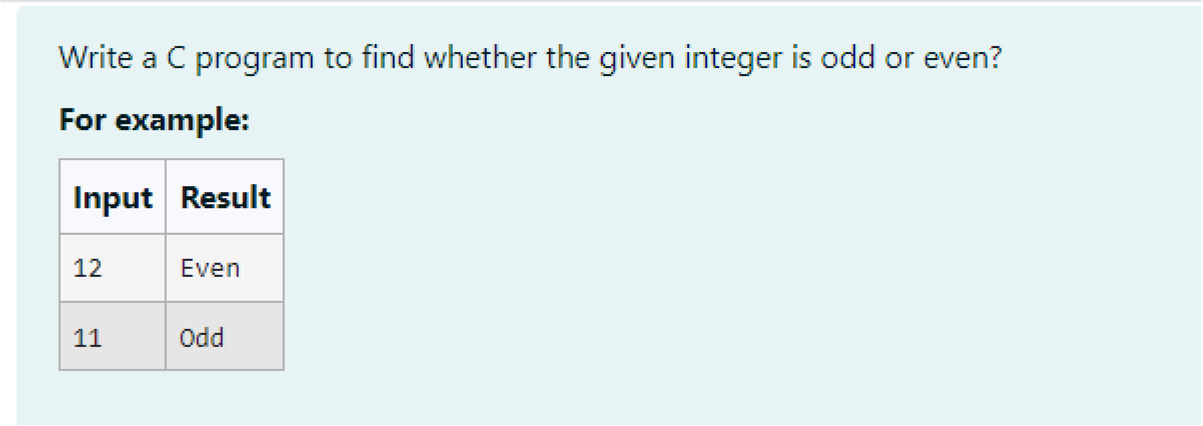
Marks for this submission: 1.00/1.00.

RESULT:

Thus the code is executed successfully and gives the expected output.

PROGRAM 9:

AIM:



Write a C program to find whether the given integer is odd or even?

For example:

Input	Result
12	Even
11	Odd

ALGORITHM:

1. Input num.

2. If num % 2 == 0:

  - Print "Even".

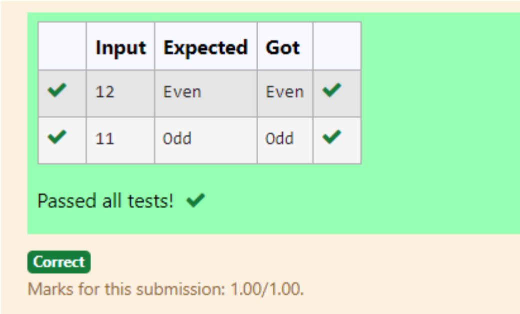
3. Else:

   - Print "Odd".

CODE:

```
#include<stdio.h>
int main()
{
    int num;
    scanf("%d",&num);
    if(num % 2 == 0)
    {
        printf("Even");
    }
    else
    {
        printf("Odd");
    }
}
```

OUTPUT:



	Input	Expected	Got	
✓	12	Even	Even	✓
✓	11	Odd	Odd	✓

Passed all tests! ✓

Correct

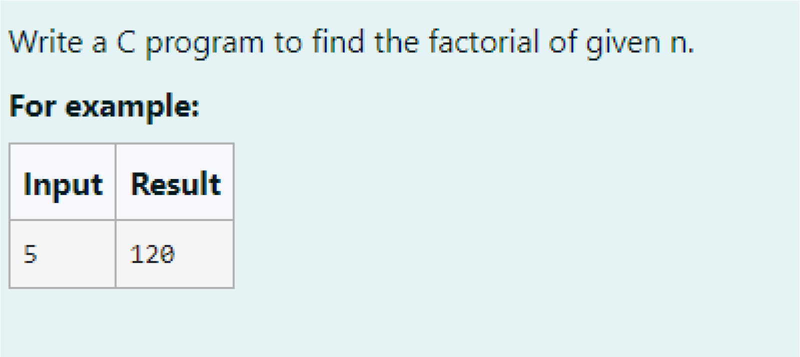
Marks for this submission: 1.00/1.00.

RESULT:

Thus the code is executed successfully and gives the expected output.

PROBLEM 10:

AIM:



Write a C program to find the factorial of given n.

For example:

Input	Result
5	120

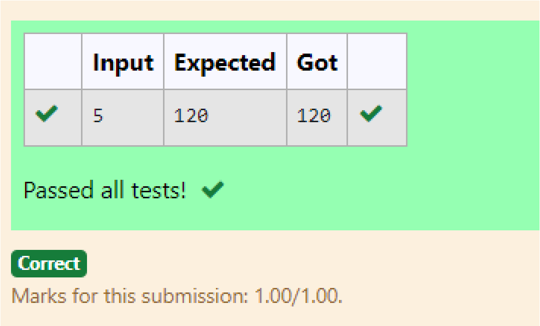
ALGORITHM:

1. Input num.
2. Initialize fact = 1.
3. Loop from i = 1 to num:
   - Multiply `fact` by `i`.
4. Print fact.

CODE:

```
#include<stdio.h>
int main()
{
    int num;
    scanf("%d",&num);
    int fact = 1;
    for(int i = 1;i <= num;i++)
    {
       fact = fact * i;
    }
    printf("%d",fact);
}
```

OUTPUT:



	Input	Expected	Got	
✓	5	120	120	✓

Passed all tests! ✓

Correct

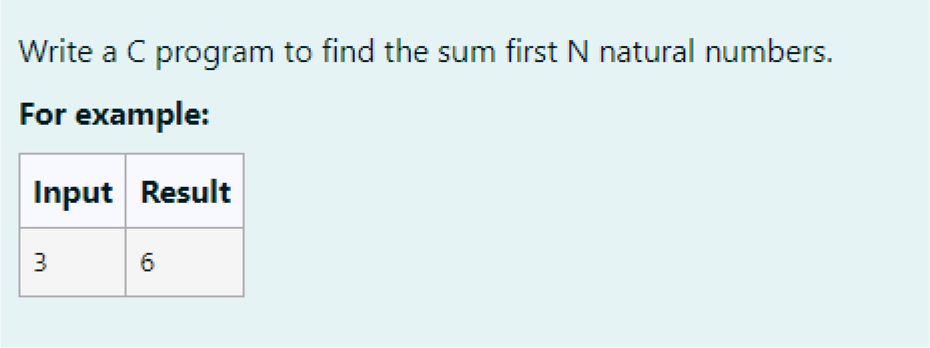
Marks for this submission: 1.00/1.00.

RESULT:

Thus the code is executed successfully and gives the expected output.

PROBLEM : 11

AIM:



Write a C program to find the sum first N natural numbers.

For example:

Input	Result
3	6

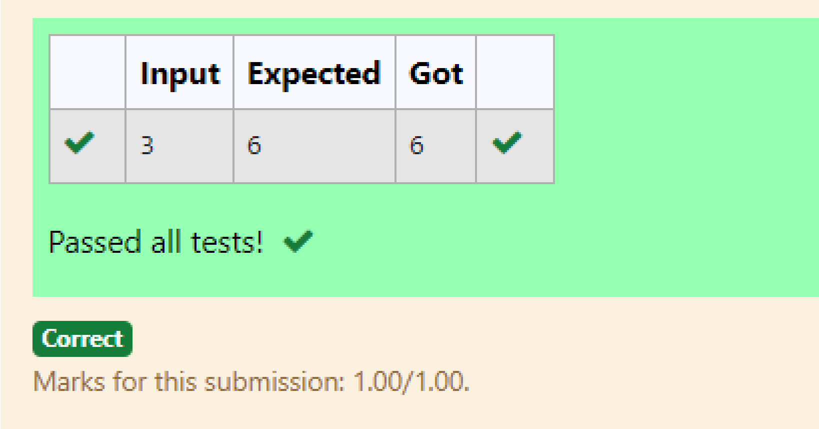
ALGORITHM:

```
1. Input a.
2. Initialize N = 0.
3. Loop from i = 1 to a:
   - Add i to N.
4. Print N.
```

CODE:

```
#include<stdio.h>
int main()
{
    int a;
    scanf("%d",&a);
    int N = 0;
    for(int i = 1;i <= a;i++)
    {
        N = N + i;
    }
    printf("%d",N);
}
```

OUTPUT:



	Input	Expected	Got	
✓	3	6	6	✓

Passed all tests! ✓

Correct

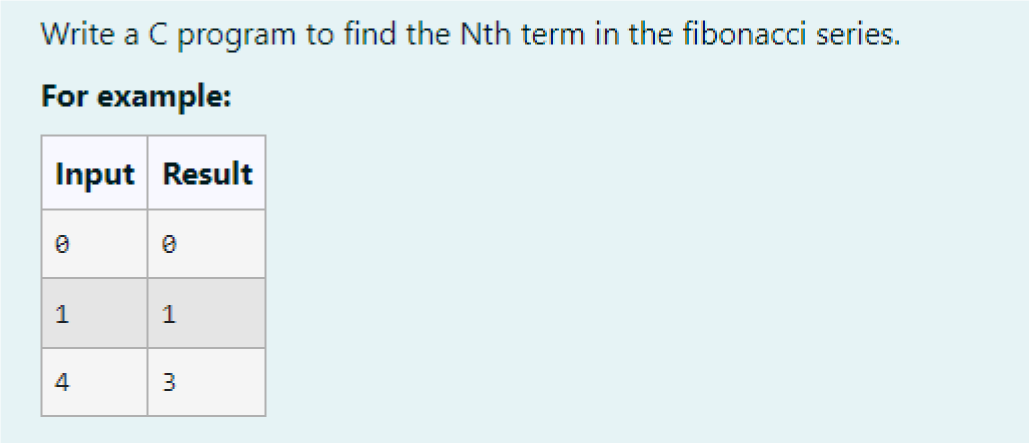
Marks for this submission: 1.00/1.00.

RESULT:

Thus the code is executed successfully and gives the expected output.

PROBLEM 12:

AIM:



Write a C program to find the Nth term in the fibonacci series.

For example:

Input	Result
0	0
1	1
4	3

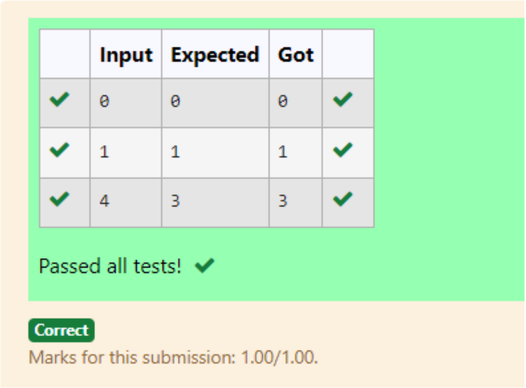
ALGORITHM:

```
1. Input n.
2. If n == 0, print 0.
3. If n == 1, print 1.
4. Else:
   - Initialize a = 1 and b = 1.
   - Loop from i = 2 to n-1:
     - Calculate c = a + b.
     - Set a = b and b = c.
5. Print c.
```

CODE:

```
#include<stdio.h>
int main()
{
    int n;
    scanf("%d",&n);
    if(n==0)
    {
        printf("%d",0);
    }
    else if(n == 1)
    {
        printf("%d",1);
    }
    else
    {
        int a,b,c;
        a=1,b=1;
        for(int i = 2;i<n;i++)
        {
            c=a+b;
            a=b;
            b=c;
        }
        printf("%d",c);
    }
}
```

OUTPUT:



	Input	Expected	Got	
✓	0	0	0	✓
✓	1	1	1	✓
✓	4	3	3	✓

Passed all tests! ✓

Correct

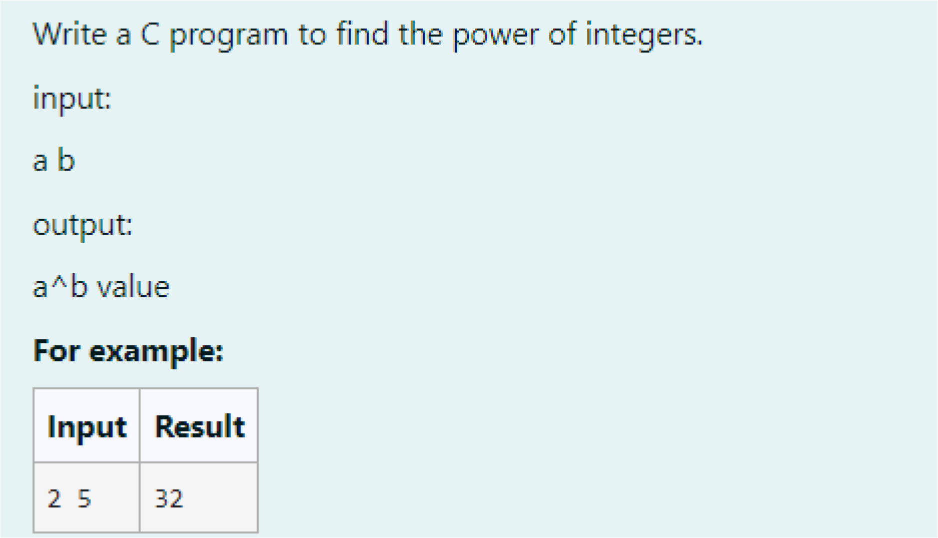
Marks for this submission: 1.00/1.00.

RESULT:

Thus the code is executed successfully and gives the expected output.

PROBLEM 13:

AIM:



Write a C program to find the power of integers.

input:

a b

output:

a^b value

For example:

Input	Result
2 5	32

ALGORITHM:

1. Input a and b.

2. Initialize c = 1.

3. Loop from i = 1 to b:

   - Multiply c by a.

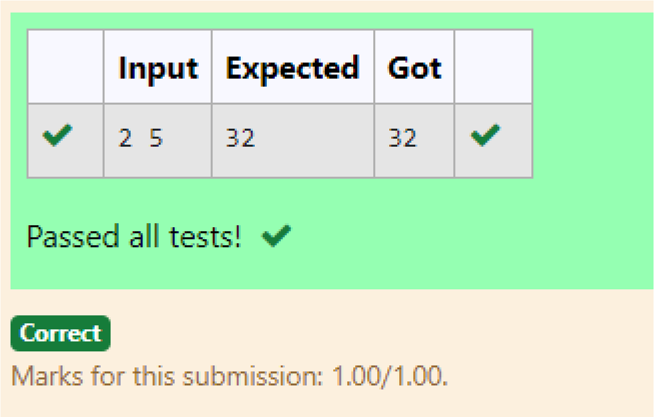
4. Print c.

CODE:

```
#include<stdio.h>
int main()
{
   int a,b,c=1,i;
   scanf("%d %d",&a,&b);
   for(i=1;i<=b;i++)
   {
      c=c*a;
   }
   printf("%d",c);
```

}

OUTPUT:



	Input	Expected	Got	
✓	2 5	32	32	✓

Passed all tests! ✓

Correct

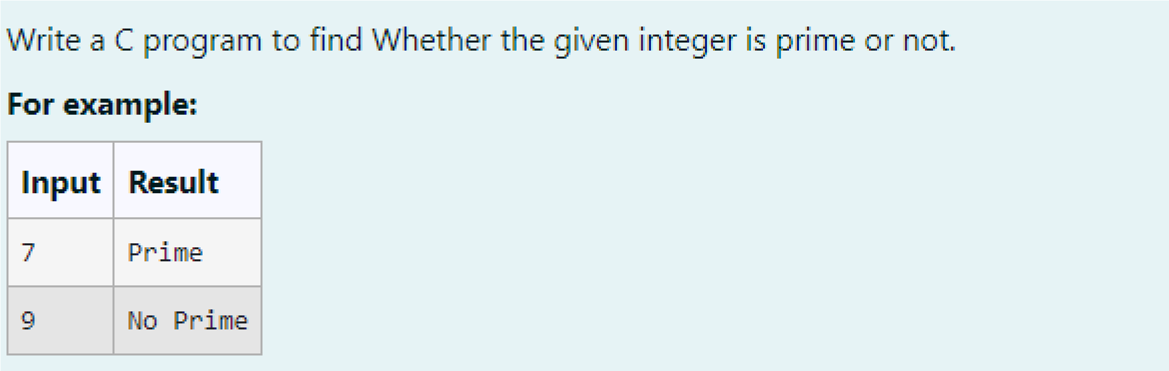
Marks for this submission: 1.00/1.00.

RESULT:

Thus the code is executed successfully and gives the expected output.

PROBLEM 14:

AIM:



Write a C program to find Whether the given integer is prime or not.

For example:

Input	Result
7	Prime
9	No Prime

ALGORITHM:

1. Input n.

2. Initialize c = 0.

3. Loop from i = 2 to n-1:

- If n % i == 0, increment c by 1.

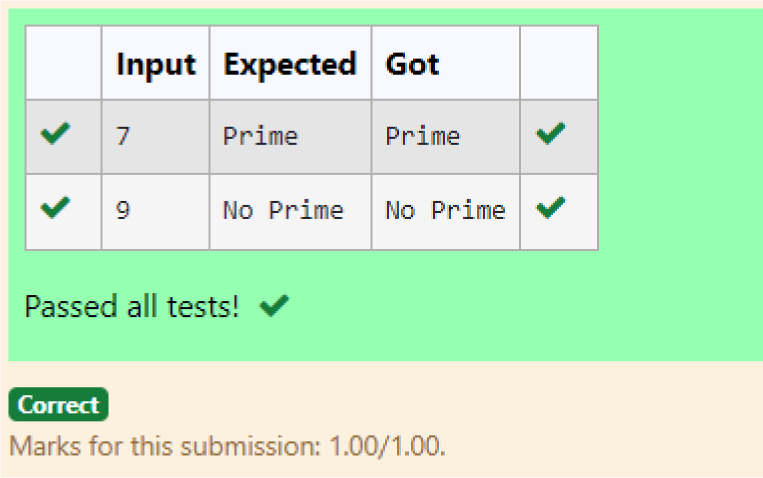
4. If c > 0, print "No Prime" (n is not prime).

5. Else, print "Prime" (n is prime).

CODE:

```
#include<stdio.h>
int main()
{
    int n,i,c=0;
    scanf("%d",&n);
    for(i=2;i<n;i++)
    {
        if(n%i==0)
        {
            c=c+1;
        }
    }
    if(c>0)
    {
        printf("No Prime");
    }
    else
    {
        printf("Prime");
    }
}
```

OUTPUT:



	Input	Expected	Got	
✓	7	Prime	Prime	✓
✓	9	No Prime	No Prime	✓

Passed all tests! ✓

Correct

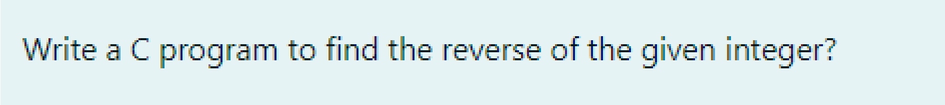
Marks for this submission: 1.00/1.00.

RESULT:

Thus the code is executed successfully and gives the expected output.

PROBLEM 15:

AIM:



Write a C program to find the reverse of the given integer?

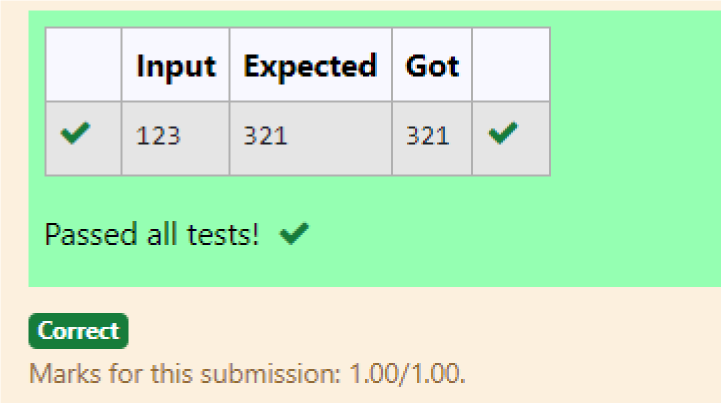
ALGORITHM:

```
1. Input the number num.
2. Initialize rev = 0.
3. While num > 0:
   - Calculate the remainder rem = num % 10.
   - Update rev = (rev * 10) + rem.
   - Update num = num / 10.
4. Output the reversed number rev.
```

CODE:

```
#include<stdio.h>
int main()
{
    int num,rev=0,rem;
    scanf("%d",&num);
    while(num>0)
    {
        rem=num%10;
        rev=(rev*10)+rem;
        num=num/10;
    }
    printf("%d",rev);
}
```

OUTPUT:



	Input	Expected	Got	
✓	123	321	321	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

RESULT:

Thus the code is executed successfully and gives the expected output.