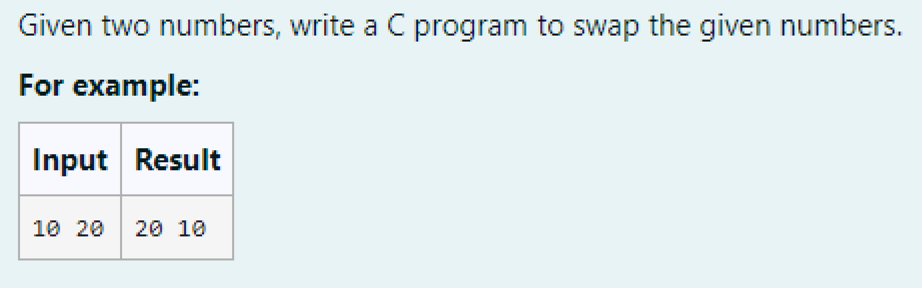
Basic C Programming Practice

1.



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

For example:

Input	Result
`10 20`	`20 10`

Algorithm:

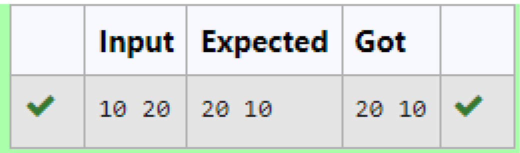
1. Start
2. Declare three integer variables: `x`, `y`, and `temp`.
3. Read the first integer value from the user and store it in `x`.
4. Read the second integer value from the user and store it in `y`.
5. Store the value of `x` in `temp`.
6. Assign the value of `y` to `x`.
7. Assign the value of `temp` to `y`.
8. Print the values of `x` and `y`.
9. End

1. Start
2. Declare three integer variables: x, y, and temp.
3. Read the first integer value from the user and store it in x.
4. Read the second integer value from the user and store it in y.
5. Store the value of x in temp.
6. Assign the value of y to x.
7. Assign the value of temp to y.
8. Print the values of x and y.
9. End

Program:

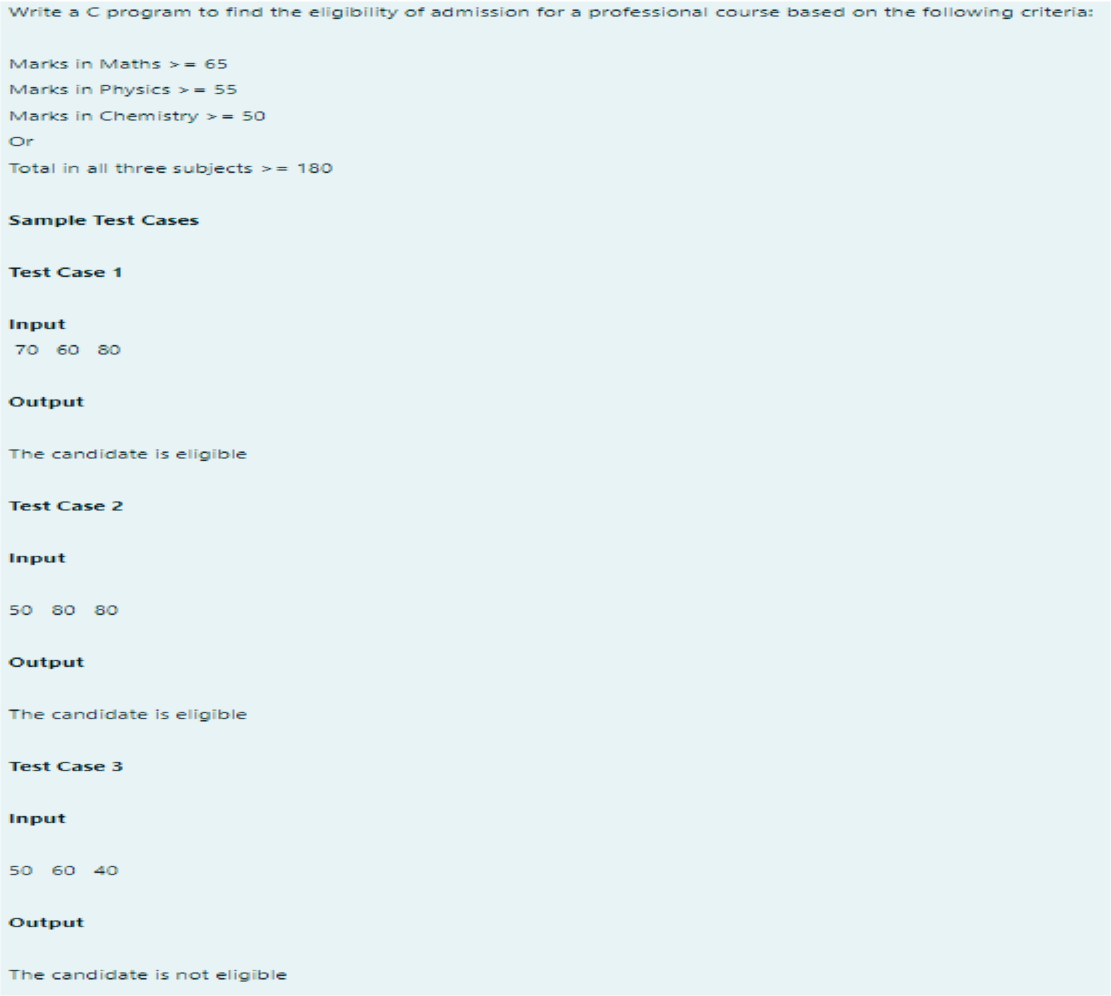
```
#include<stdio.h>
```

```
 int main()
{
int x,y,temp;
 scanf("%d",&x);
 scanf("%d",&y);
 temp=x;
 x=y;
 y=temp;
 printf("%d %d",x,y);
}
```



	Input	Expected	Got	
✓	10 20	20 10	20 10	✓

2.



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

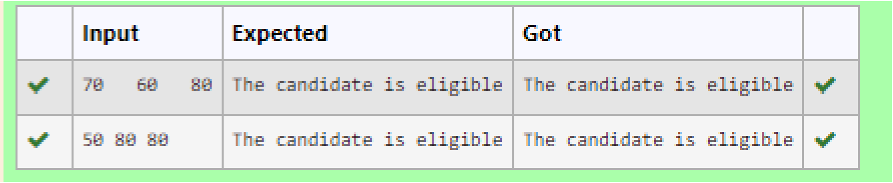
```
The candidate is not eligible
```

Algorithm:

1. Start
2. Declare three integer variables p, c, and m.
3. Read the values of p, c, and m from the user.
4. If the conditions is true:
5. Print "The candidate is eligible" Else:
6. Print "The candidate is not eligible".
7. End

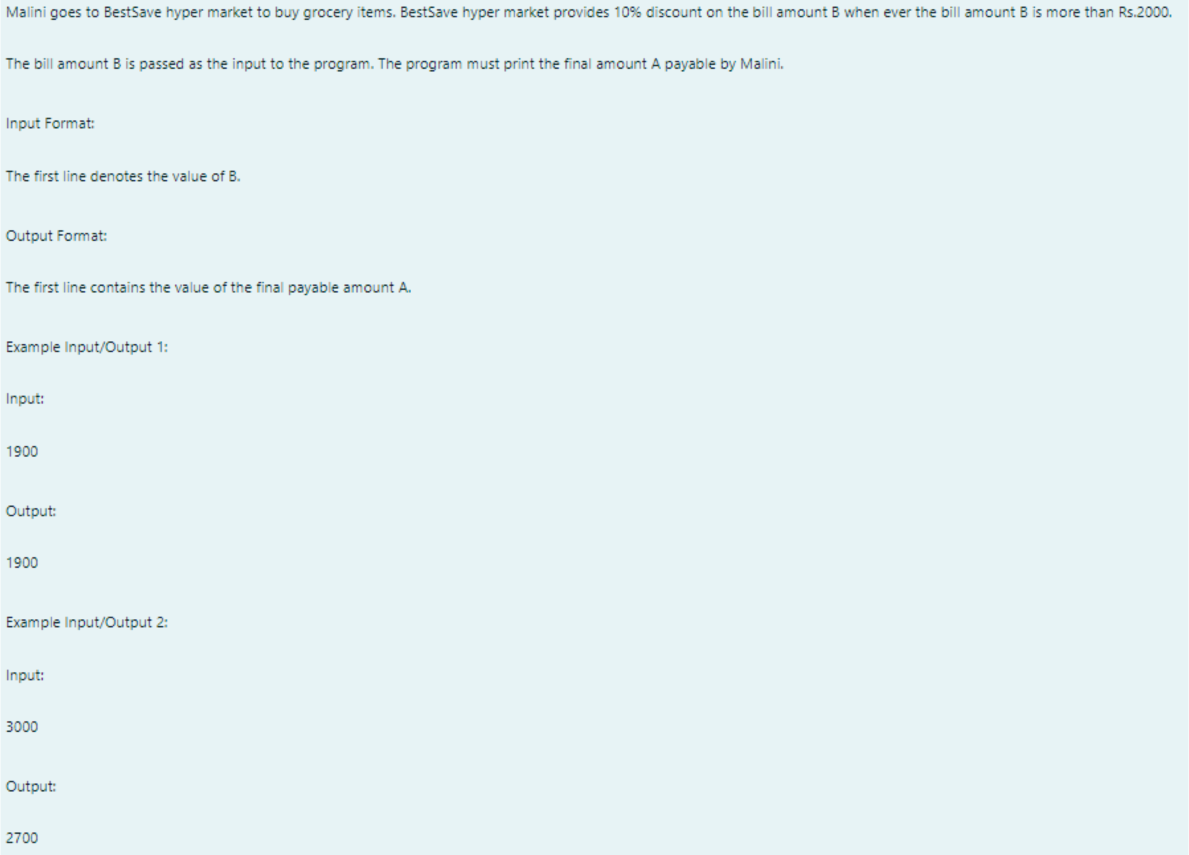
Program:

```
#include<stdio.h>

int main()

{

   int p,c,m;

   scanf("%d %d %d",&p,&c,&m);

   if((m>=65 && p>=55 && c>=50)||(m+p+c>=180))

   {

      printf("The candidate is eligible");

   }

   else

   {

      printf("The candidate is not eligible");

   }

}
```



	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	✓

3.



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. Start

2. Declare an integer variable b.

3. Read the value of b from the user.

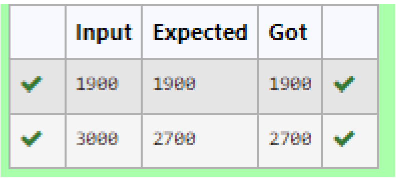
4. If b > 2000, calculate the discount dis = (b * 10) / 100 and the final amount f = b - dis. Print f`.

5. Otherwise, print b

6. End

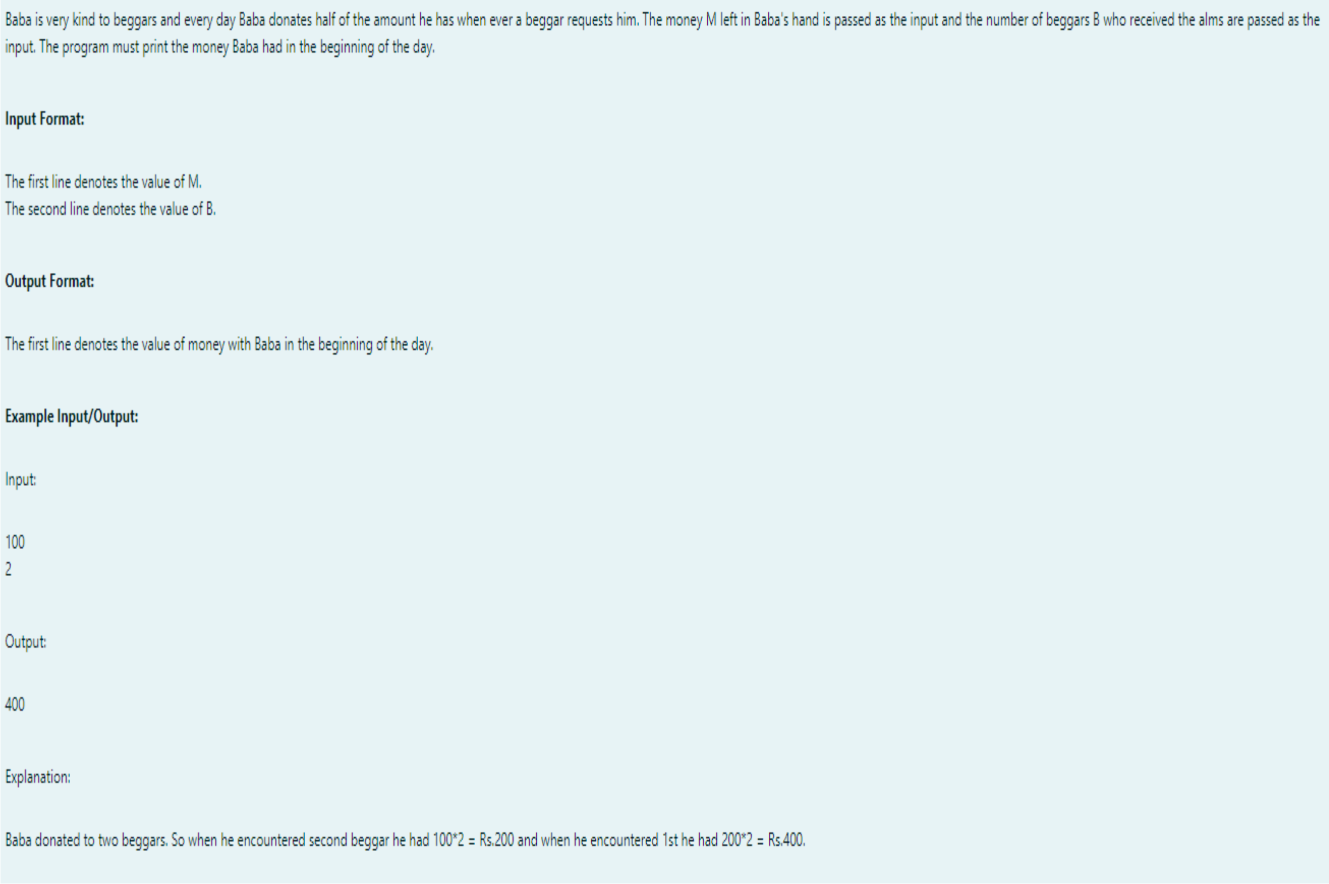
Program:

```
#include<stdio.h>
int main()
{
    int b,f;
    scanf("%d",&b);
    if(b>2000)
    {
        int dis;
        dis=(b*10)/100;
        f=b-dis;
        printf("%d",f);
    }
    else
    {
        printf("%d",b);
    }
}
```



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

4.



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:

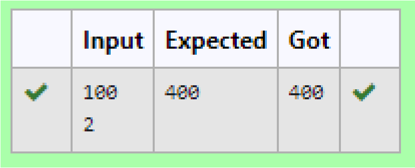
Baba donated to two beggars. So when he encountered second beggar he had 100*2 = Rs.200 and when he encountered 1st he had 200*2 = Rs.400.

Algorithm:

1. Start
2. Declare two integer variables m and b.
3. Read the values of m and b from the user.
4. While b != 0, multiply m by 2 and decrement b by 1.
5. Print the value of m.
6. End

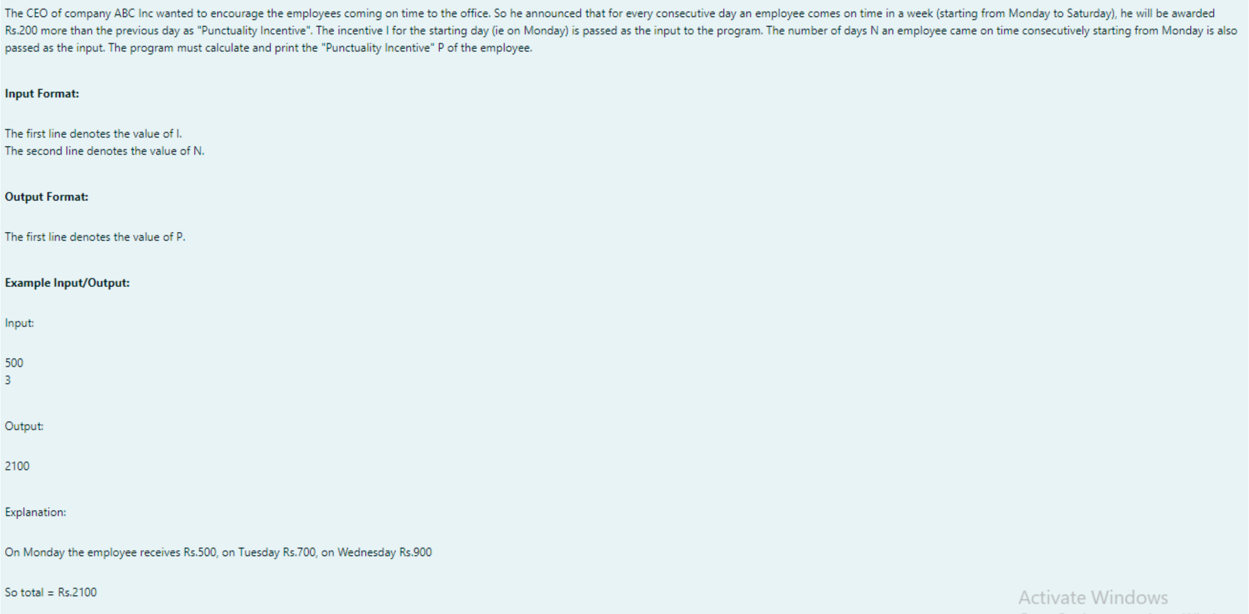
Program:

```
#include<stdio.h>

int main()

{
    int m,b;
    scanf("%d %d",&m,&b);
    while(b!=0)
    {
        m=m*2;
        b--;
    }
    printf("%d",m);
}
```



	Input	Expected	Got	
✓	100 2	400	400	✓

5.



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

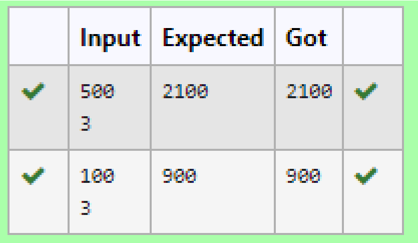
Algorithm:

1. Start
2. Declare three integer variables i, d, and s.
3. Read the values of i and d from the user.
4. Set s = i.
5. Print the value of s.
6. End

Program:

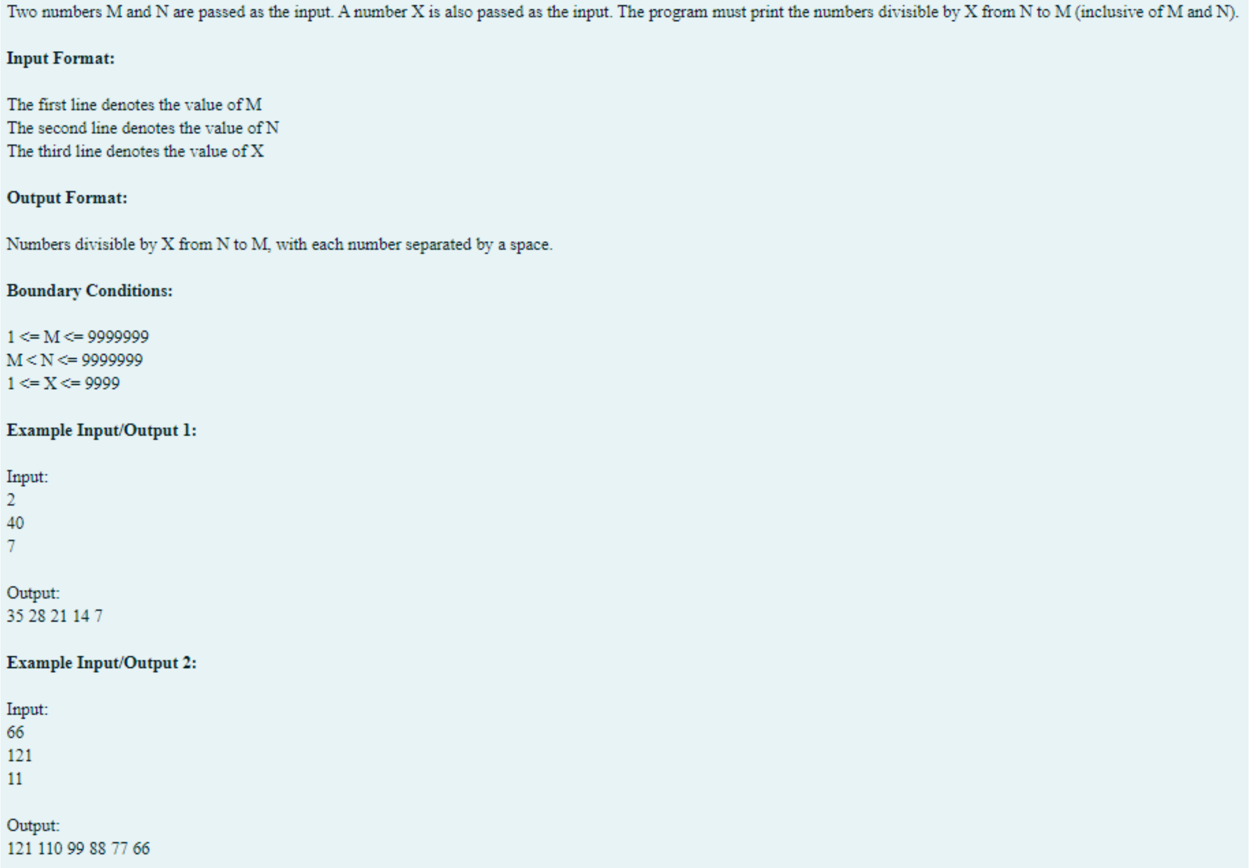
```
#include<stdio.h>

int main()

{

   int i,d,s;

   scanf("%d %d",&i,&d);
```

```
    s=i;

    while(d!=1)

    {

        i=i+200;

        s=s+i;

        d--;

    }

    printf("%d",s);

}
```



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

6.



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
```

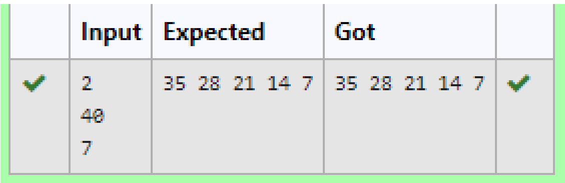
Algorithm:

1. Start
2. Declare three integer variables m, n, and x.
3. Read the values of m, n, and x from the user.
4. Set i = n.
5. While i >= m, do the following:
6. If i % x == 0, print i.
7. Decrement i by 1.
8. End

Program:

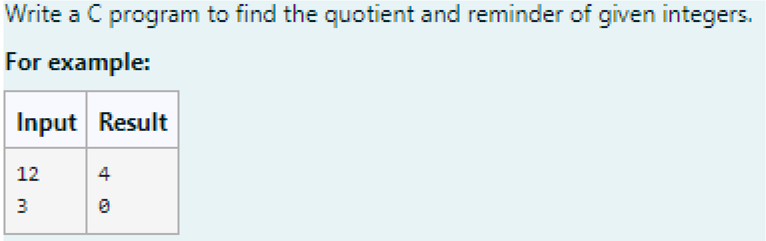
```
#include<stdio.h>
```

```
int main()
{
    int m,n,x;
    scanf("%d %d %d",&m,&n,&x);
    int i=n;
    while(i>=m)
    {
        if(i%x==0)
        {
            printf("%d ",i);
        }
        i--;
    }
}
```



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

7.



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

For example:

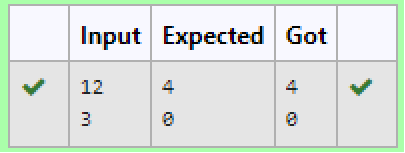
Input	Result
12 3	4 0

Algorithm:

1. Start
2. Declare two integer variables a and d.
3. Read the value of a from the user.
4. Read the value of d from the user.
5. Calculate q = a / d and r = a % d.
6. Print the value of q.
7. Print the value of r.
8. End

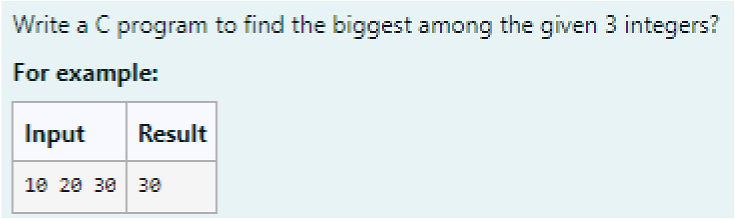
Program:

```
#include<stdio.h>

int main()

{

   int a,d;

   scanf("%d",&a);

   scanf("%d",&d);

   int q=a/d;

   int r=a%d;

   printf("%d",q);

   printf("\n");

   printf("%d",r);

}
```



	Input	Expected	Got	
✓	12 3	4 0	4 0	✓

8.



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

For example:

Input	Result
10 20 30	30

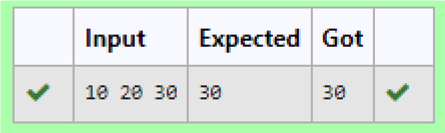
Algorithm:

1. Start
2. Declare three integer variables a, b, and c.
3. Read the values of a, b, and c from the user.
4. If a is greater than both b and c, print a.
5. Else if b is greater than both a and c, print b.
6. Else, print c.
7. End

Program:

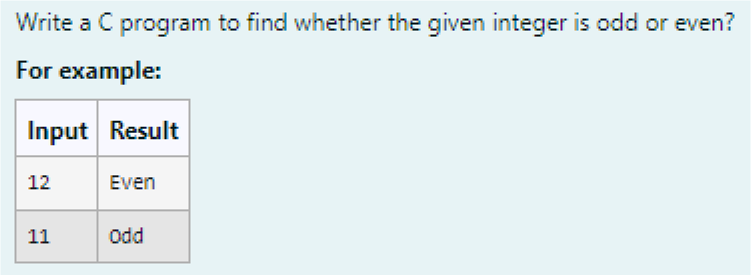
```
#include<stdio.h>

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);
 }
 }
```



	Input	Expected	Got	
✓	10 20 30	30	30	✓

9.



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

For example:

Input	Result
12	Even
11	Odd

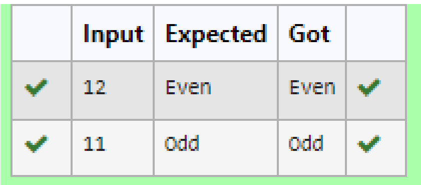
Algorithm:

1. Start
2. Declare an integer variable a.
3. Read the value of a from the user.
4. If a is divisible by 2 print "Even".
5. Else, print "Odd".
6. End

Program:

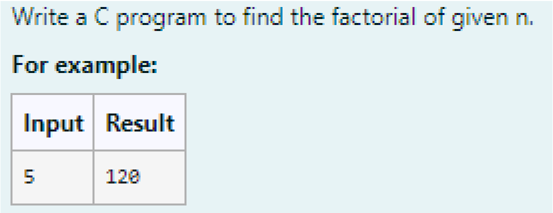
```
#include<stdio.h>

int main()
```

```
{
  int a;
  scanf("%d",&a);
  if(a%2==0)
  {
    printf("Even");
  }
  else
  {
    printf("Odd");
  }}
```



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

10.



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

For example:

Input	Result
5	120

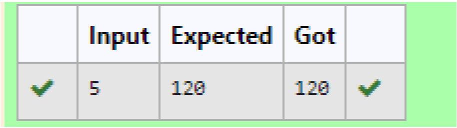
Algorithm:

1. Start
2. Declare two integer variables a and x.
3. Read the value of a from the user.
4. Set x = a.
5. While x != 1
6. Decrement x by 1.
7. Multiply a by x.
8. Print the value of a.

9. End

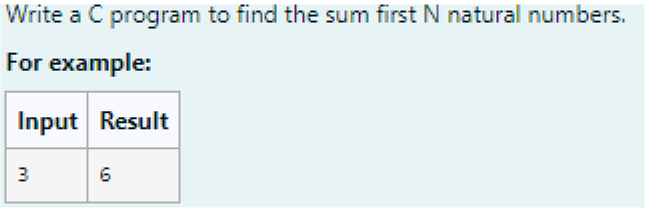
Program:

```
#include<stdio.h>
int main()
{
   int a,x;
   scanf("%d",&a);
   x=a;
   while(x!=1)
   {
      x--;
      a=a*x;
   }
   printf("%d",a);
}
```



	Input	Expected	Got	
✓	5	120	120	✓

11.



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

For example:

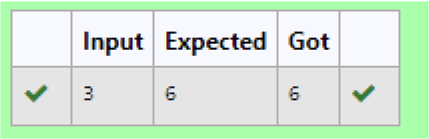
Input	Result
3	6

Algorithm:

1. Start
2. Declare two integer variables a and x.
3. Read the value of a from the user.
4. While x != 1, do the following:
5. Decrement x by 1.
6. Multiply a by x.
7. Print the value of a.
8. End

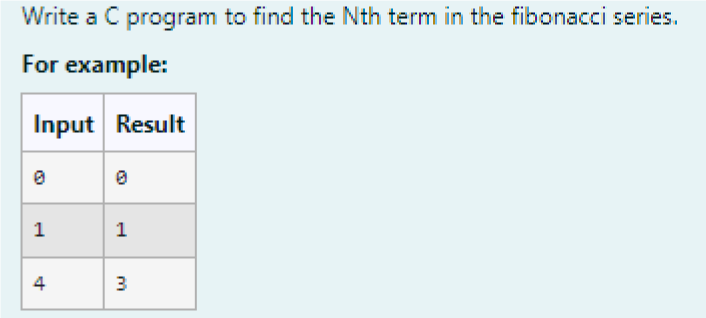
Program:

```
#include<stdio.h>

int main()

{

   int a,b=0;

   scanf("%d",&a);

   while(a!=0)

   {

      b+=a;

      a--;

   }

   printf("%d",b);

}
```



	Input	Expected	Got	
✓	3	6	6	✓

12.



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

For example:

Input	Result
0	0
1	1
4	3

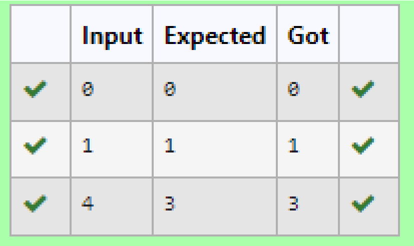
Algorithm:

1. Start
2. function fib(n) that returns the nth Fibonacci number:
3. If n <= 1, return n.
4. Otherwise, return fib(n-1) + fib(n-2).
5. Declare an integer variable n.
6. Read the value of n from the user.
7. Calling the fib(n) function and print the result.
8. End

Program:

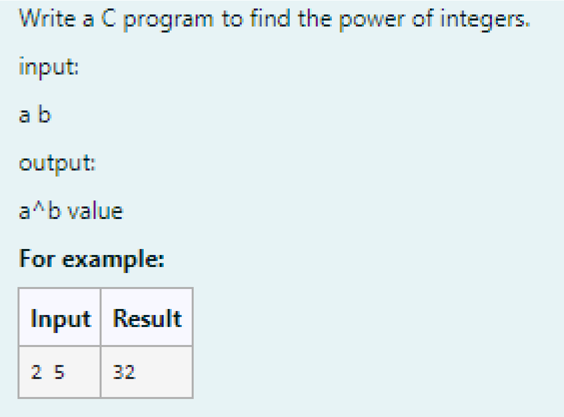
```
#include<stdio.h>

int fib(int n){

   if(n<=1){

      return n;

   }

   else{

      return fib(n-1)+fib(n-2);

   }

}

int main()

{

   int n;

   scanf("%d",&n);

   printf("%d",fib(n));
```

```
    return 0;

}
```



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

13.



Write a C program to find the power of integers.

input:

a b

output:

a^b value

For example:

Input	Result
2 5	32

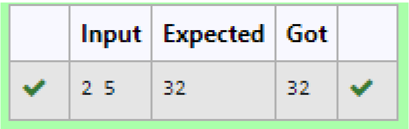
Algorithm:

1. Start
2. Declare two integer variables a and b.
3. Read the values of a and b from the user.
4. Declare two integer variables i = 0 and p = 1.
5. While i < b
6. Multiply p by a.
7. Increment i by 1.
8. Print the value of p.
9. End

Program:

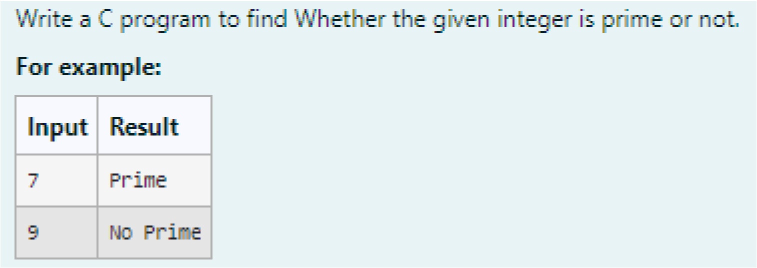
```
#include<stdio.h>

int main()

{

    int a,b;
```

```
   scanf("%d",&a);

   scanf("%d",&b);

   int i=0,p=1;

   while(i<b){

     p=p*a;

     i++;

   }

   printf("%d",p);

}
```



	Input	Expected	Got	
✓	2 5	32	32	✓

14.



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

For example:

Input	Result
7	Prime
9	No Prime

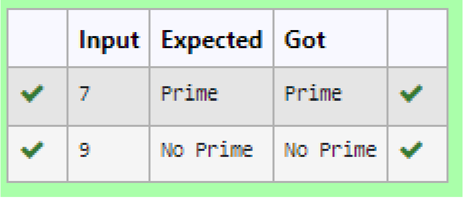
Algorithm:

1. Start
2. Declare two integer n and flag.
3. Read the value n from the user.
4. If n % i == 0, set flag = 1
5. Else, set flag = 0.
6. If flag == 0, print "Prime".
7. Otherwise, print "No Prime".
8. End

Program:

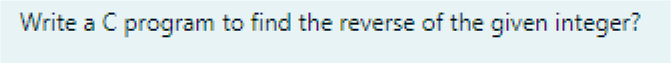
```
#include<stdio.h>
```

```
int main()
{
   int n,flag;
   scanf("%d",&n);
   for (int i=2;i<n;i++){
      if(n%i==0){
         flag=1;
         break;
      }
      else{
         flag=0;
      }
   }
   if(flag==0){
      printf("Prime");
   }
   else{
      printf("No Prime");
   }


}
```



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

15.



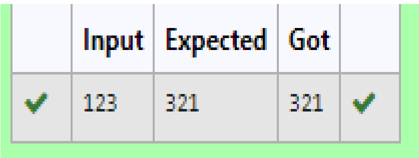
Write a C program to find the reverse of the given integer?

Algorithm:

1. Start
2. Declare three integer variables n, rem, and rev
3. Read the value of n from the user.
4. While n != 0
5. Calculate rem = n % 10.
6. Update rev = rev * 10 + rem.
7. Update n = n / 10.
8. Print the value of rev.
9. End

Program:

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



	Input	Expected	Got	
✓	123	321	321	✓