Ex. No: 1 **Date: 12.08.24**

Register No.: 230701386 **Name: R. YASHVINTHINI**

Basic C Programming

1.a.

Aim:

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

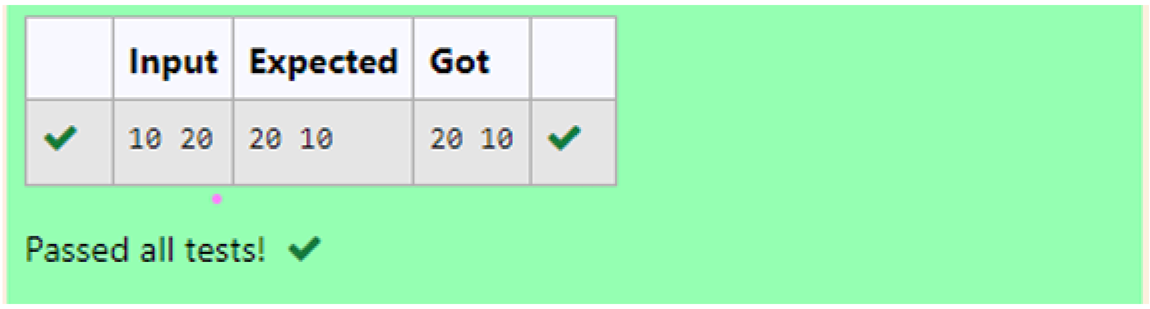
Algorithm:

1. declare n1, n2, temp as integer
2. read n1,n2
3. copy n1 into temp
4. update value of n2 to n1
5. update value of n1 to temp

Program:

```
#include <stdio.h>;

 int main()

{

        int n1,n2,temp;

       scanf("%d",&n1);

        scanf("%d",&n2);

       temp=n1;

       n1=n2;

       n2=temp;

       printf("%d %d",n1,n2);

 }
```

Output:



	Input	Expected	Got	
✓	10 20	20 10	20 10	✓

Passed all tests! ✓

PROGRAM 2:

AIM:

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

Marks in Math >= 65

Marks in Physics >= 55

[or]

Total in all subjects >=180

Marks in Chemistry >= 50

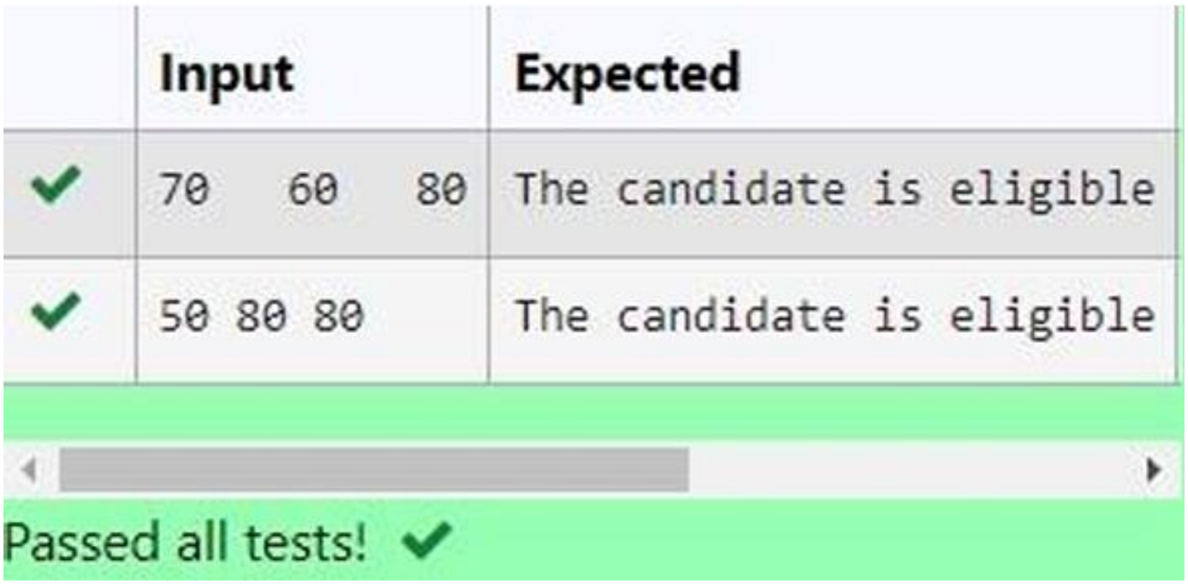
ALGORITHM:

1. Initialize Variables
2. Input Marks
3. If marks in Math are greater than or equal to 65, marks in Physics are greater than or equal to 55, and marks in Chemistry are greater than or equal to 50, then the candidate is eligible.
4. Otherwise, if the total marks in all three subjects are greater than or equal to 180, then the candidate is eligible.
5. If neither condition is met, the candidate is not eligible.
6. Output Result

PROGRAM:

```
#include <stdio.h>


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


  if (m >= 65 && p >= 55 && c >= 50) {
    printf("The candidate is eligible\n");
    } else if (m + p + c >= 180) {
        printf("The candidate is eligible\n");
    } else {
        printf("The candidate is not eligible\n");
    }

    return 0;
}
```

OUTPUT:



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

Passed all tests! ✓

RESULT:

Thus, the program is executed successfully.

PROGRAM 3:

AIM:

Malini goes to Best save hyper market to buy grocery items. Best save hypermarket provides 10% discount on the bill amount B whenever the bill amount B is more than Rs. 2000. The bill amount B is passed as the input to the program and it must print the final amount payable by Malini.

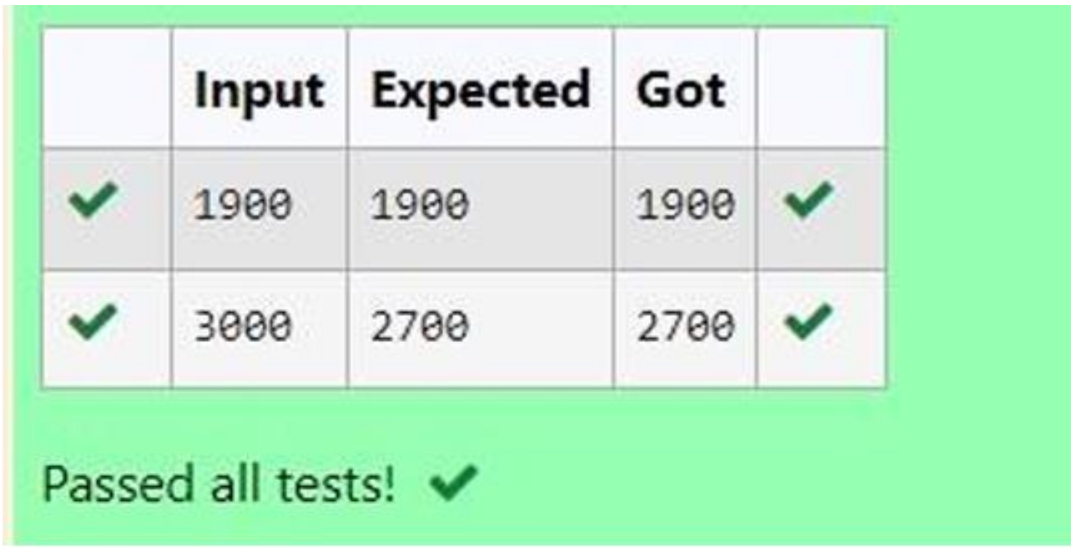
ALGORITHM:

1. Initialize Variables
2. Input Bill Amount
3. If the bill amount B is greater than Rs. 2000, calculate the discount as 10% of B and subtract it from the original bill amount.
4. Otherwise, the bill amount remains the same.
5. Output Final Amount

PROGRAM:

```c
#include <stdio.h>


int main() {
  float B, disc;
  scanf("%f", &B);


  if (B > 2000) {
    disc = B * 0.10;
    B = B - disc;
  }


  printf("%.2f\n", B);
  return 0;
}
```

OUTPUT:



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

Passed all tests! ✓

RESULT:

Thus, the program is executed successfully.

PROGRAM 4:

AIM:

Baba is very kind to beggars and every day Baba donates half of the amount he has whenever 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 at the beginning of the day.

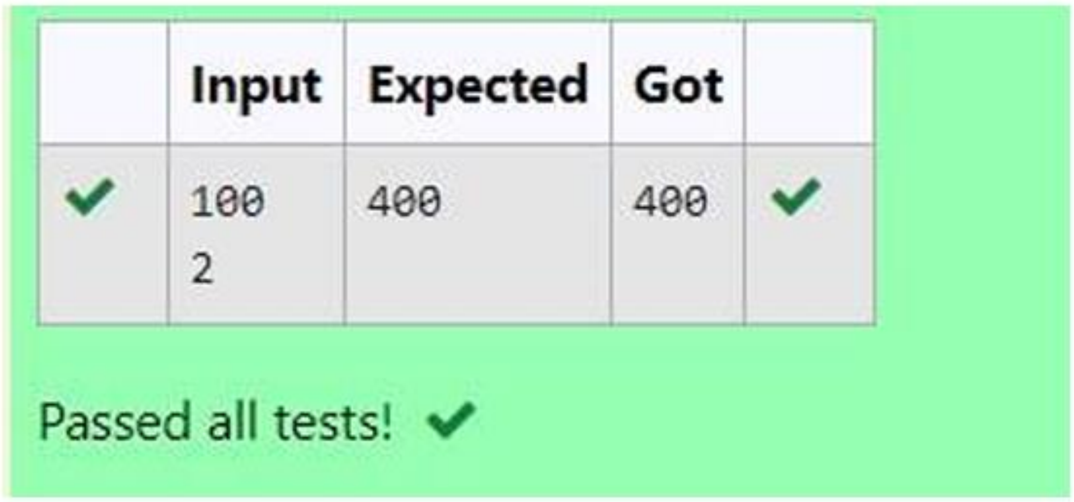
ALGORITHM:

1. Initialize Variables
2. Input Values
3. Initialize a variable initial_amount with the value of m.
4. For each beggar (from 1 to B), multiply initial_amount by 2 (since Baba donates half of his money each time).
5. Output Result

PROGRAM:

```
#include <stdio.h>


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


  int initial_amount = m;
  for (int i = 0; i < B; i++) {
    initial_amount *= 2;
  }


  printf("%d\n", initial_amount);
  return 0;
}
```

OUTPUT:



	Input	Expected	Got	
✓	100 2	400	400	✓

Passed all tests! ✓

RESULT:

Thus, the program is executed successfully.

PROGRAM 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 [starting from Monday through Saturday] he will be awarded Rs. 200 more than the previous day as “Punctuality incentive”. Incentive for starting day is passed as input and the number of days N is also passed. The program is to calculate the “Punctuality incentive” P of the employee.

ALGORITHM:

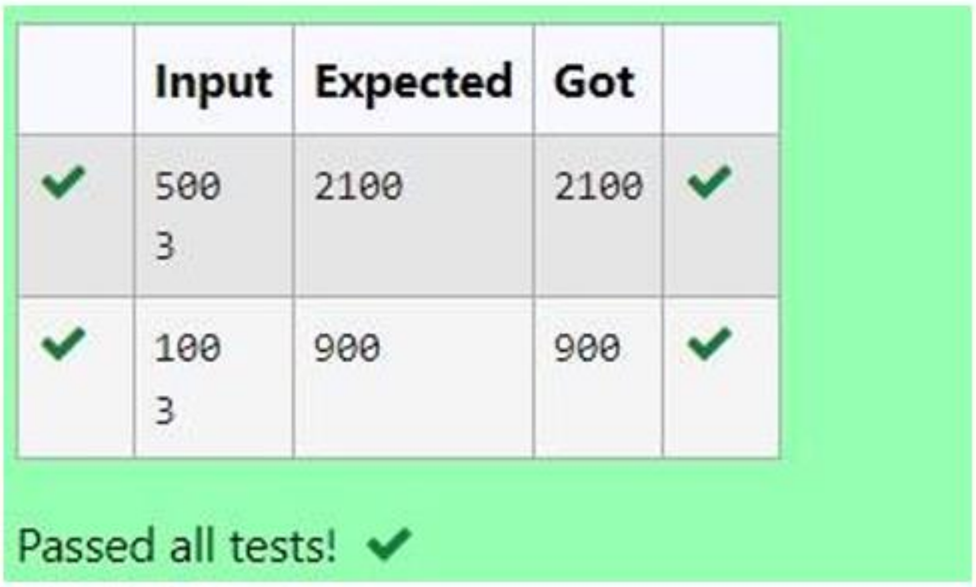
1. Initialize Variables
2. Input Values
3. Initialize sum with the value of the initial incentive.
4. For each day from 1 to n-1, increment the incentive by Rs. 200 and add it to sum.

 Output Result

PROGRAM:

```
#include <stdio.h>


int main() {

  int i, n, sum;


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


  sum = i;

  for (int j = 1; j < n; j++) {

    i += 200;

    sum += i;

  }

  return 0;

}
```

OUTPUT:



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

Passed all tests! ✓

RESULT:

Thus, the program is executed successfully.

PROGRAM 6:

AIM:

Two numbers a and b are passed as the input. A number x is also passed as the input. The program must print the numbers divisible by x from b to a range inclusive of a and b.

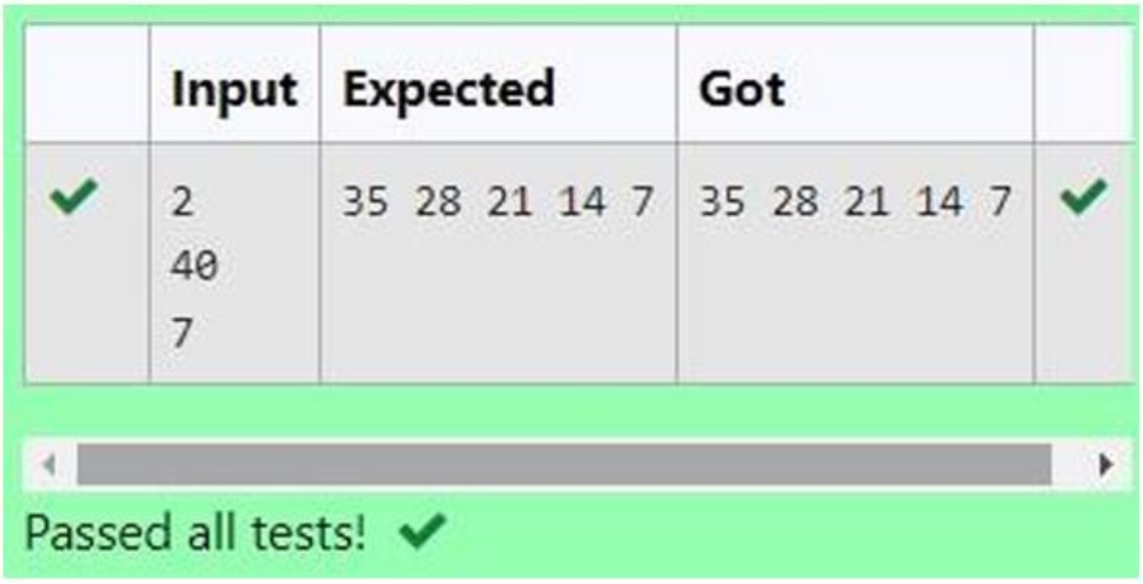
ALGORITHM:

1. Initialize Variables
2. Input Values
3. Use a for loop to iterate from b to a (inclusive).
4. For each number in the range, check if it is divisible by x.
5. If it is divisible, print the number.
6. Output result

PROGRAM:

```
#include <stdio.h>


int main() {
  int a, b, x;
  scanf("%d %d %d", &a, &b, &x);


  printf("\n", x, b, a);
  for (int i = b; i <= a; i++) {
    if (i % x == 0) {
      printf("%d ", i);
    }
  }
  printf("\n");


  return 0;
}
```

OUTPUT:



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

Passed all tests! ✓

RESULT: Thus, the program is executed successfully.

PROGRAM 7:

AIM:

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

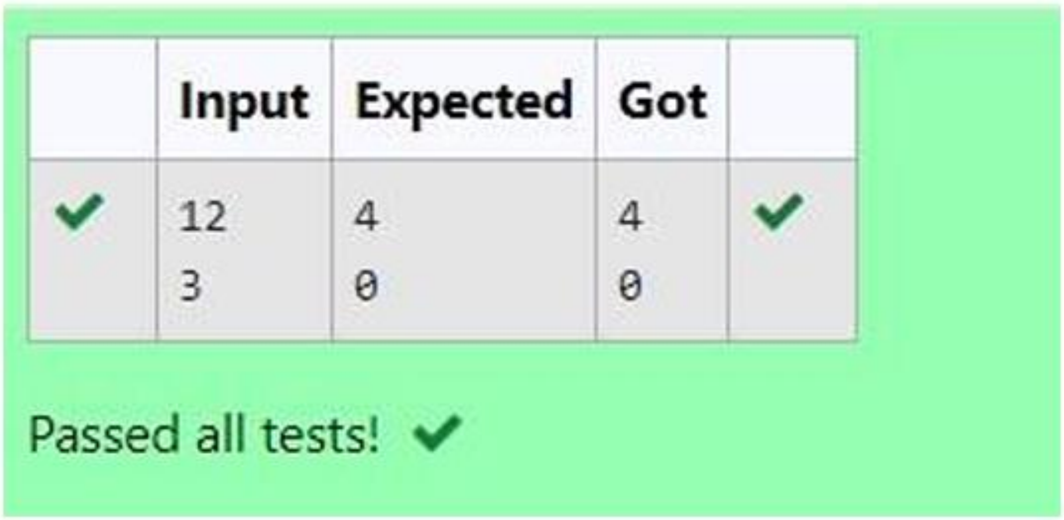
ALGORITHM:

```
1:Initialize the 2 numbers a and b.
2: Take an input for a and b from the user.
3: Display a/b and a%b.
```

PROGRAM:

```
#include int main() {
int a,b;
 scanf("%d%d",&a,&b);
 printf("%d\n",a/b);
 printf("%d",a%b);
 }
```

OUTPUT:



	Input	Expected	Got	
✓	12 3	4 0	4 0	✓

Passed all tests! ✓

RESULT:

Thus, the program is executed successfully.

PROGRAM 8:

AIM:

Write a program to find the biggest number out of the 3 given integers.

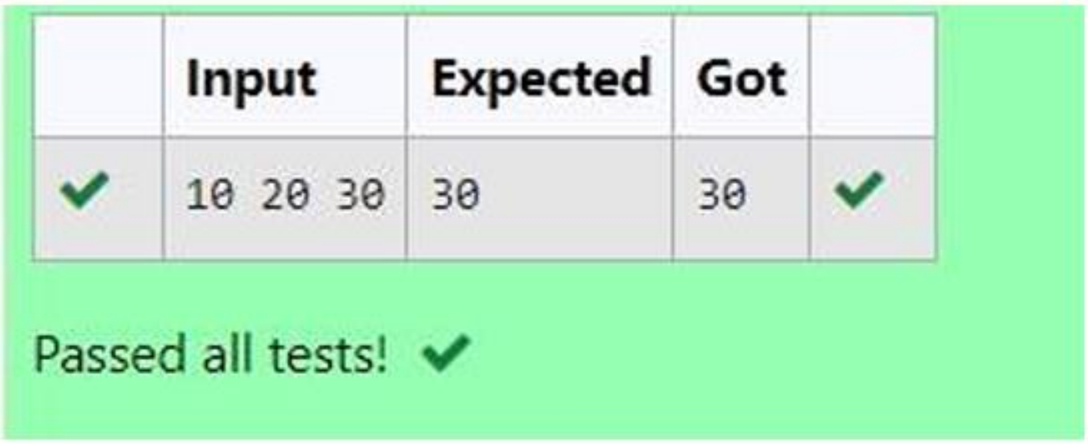
ALGORITHM:

1. Initialize Variables
2. Input Values
3. If a is greater than or equal to both b and c, then a is the largest.
4. Otherwise, if b is greater than or equal to both a and c, then b is the largest.
5. Otherwise, c is the largest.
6. Output Result

PROGRAM:

```
#include <stdio.h>


int main() {
  int a, b, c;
  scanf("%d %d %d", &a, &b, &c);


  if (a >= b && a >= c) {
    printf("%d\n", a);
  } else if (b >= a && b >= c) {
    printf("%d\n", b);
  } else {
    printf("%d\n", c);
  }


  return 0;
}
```

OUTPUT:



	Input	Expected	Got	
✓	`10 20 30`	`30`	`30`	✓

Passed all tests! ✓

RESULT:

Thus, the program is executed successfully.

PROGRAM 9:

AIM:

Write a C program to find whether a number is even or odd

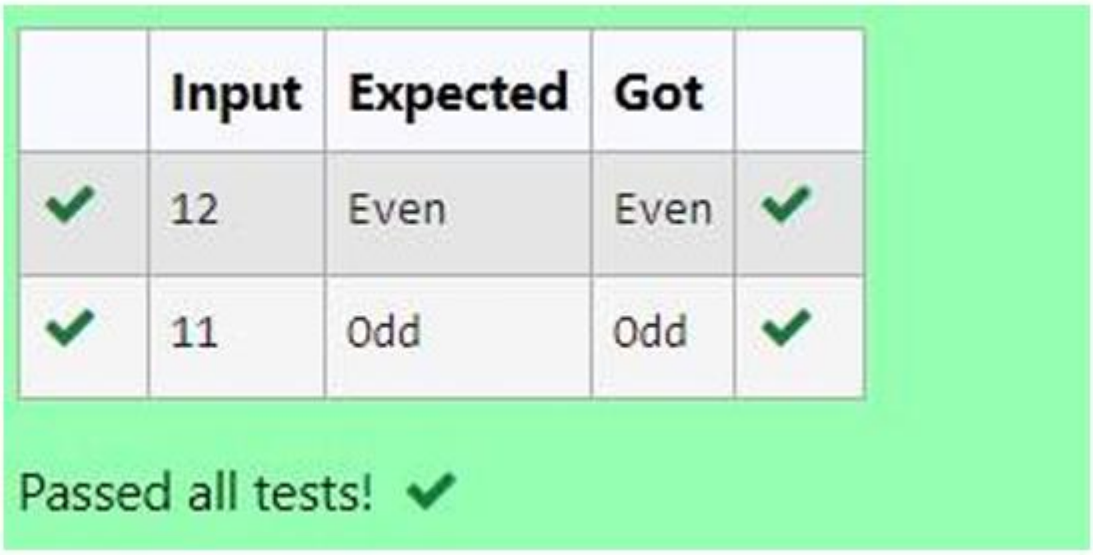
ALGORITHM:

1. Initialize Variable
2. Input Value
3. If n % 2 == 0, the number is even.
4. Otherwise, the number is odd.
5. Output Result

PROGRAM:

```
#include <stdio.h>


int main() {

  int n;

  scanf("%d", &n);


  if (n % 2 == 0) {

    printf(“Even”);

  } else {

    printf (“Odd");

  }


  return 0;

}
```

OUTPUT:



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

Passed all tests! ✓

RESULT:

Thus, the program is executed successfully.

PROGRAM 10:

AIM:

Write a C program to find the factorial of a number N.

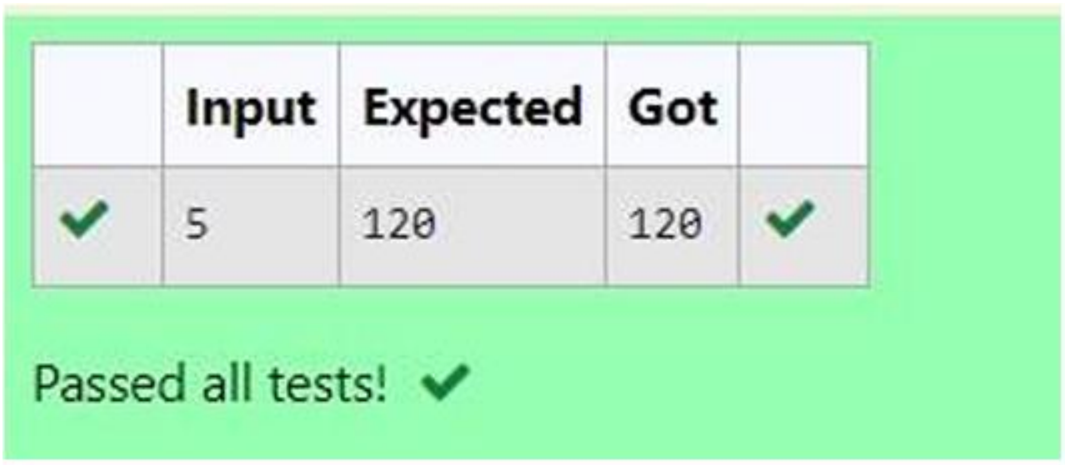
ALGORITHM:

1. Initialize Variables
2. Input Value
3. Use a for loop to iterate from 1 to NN.
4. Multiply the factorial result by the loop counter ii in each iteration.
5. Output Result

PROGRAM:

```
#include <stdio.h>


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



  if (N < 0) {
    printf("\n");
  } else {
    for (i = 1; i <= N; ++i) {
      factorial *= i;
    }
    printf("%d\n",  factorial);
  }

  return 0;
}
```

OUTPUT:



	Input	Expected	Got	
✓	5	120	120	✓

Passed all tests! ✓

RESULT:

Thus, the program is executed successfully.

PROGRAM 11:

AIM:

Write a C program to find sum of first N natural.

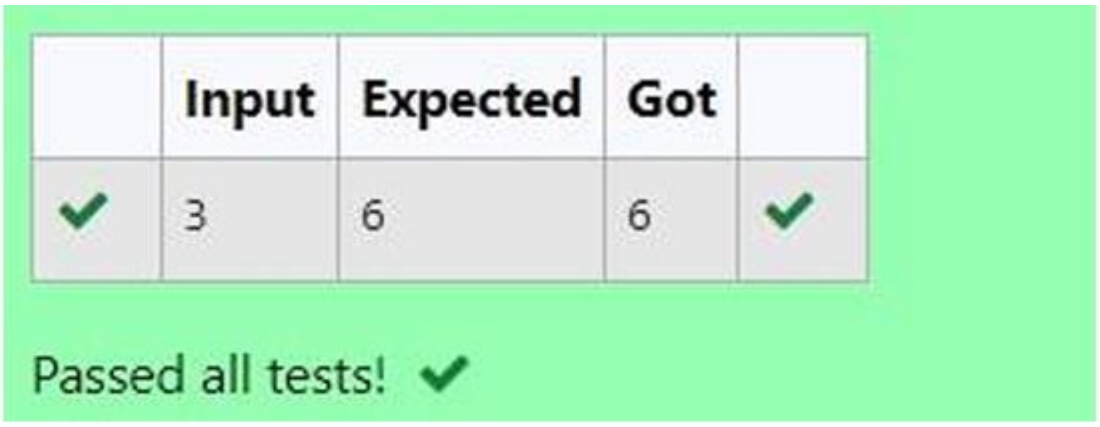
ALGORITHM:

1. Initialize Variables
2. Input Value
3. Use a for loop to iterate from 1 to NN.
4. Add each number to the sum.
5. Output Result

PROGRAM:

```
#include <stdio.h>


int main() {
   int N, sum = 0;
   scanf("%d", &N);


   for (int i = 1; i <= N; ++i) {
      sum += i;
   }


   printf("%d\n", N, sum);
   return 0;
}
```

OUTPUT:



	Input	Expected	Got	
✓	3	6	6	✓

Passed all tests! ✓

RESULT:

Thus, the program is executed successfully.

12:

AIM:

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

ALGORITHM:

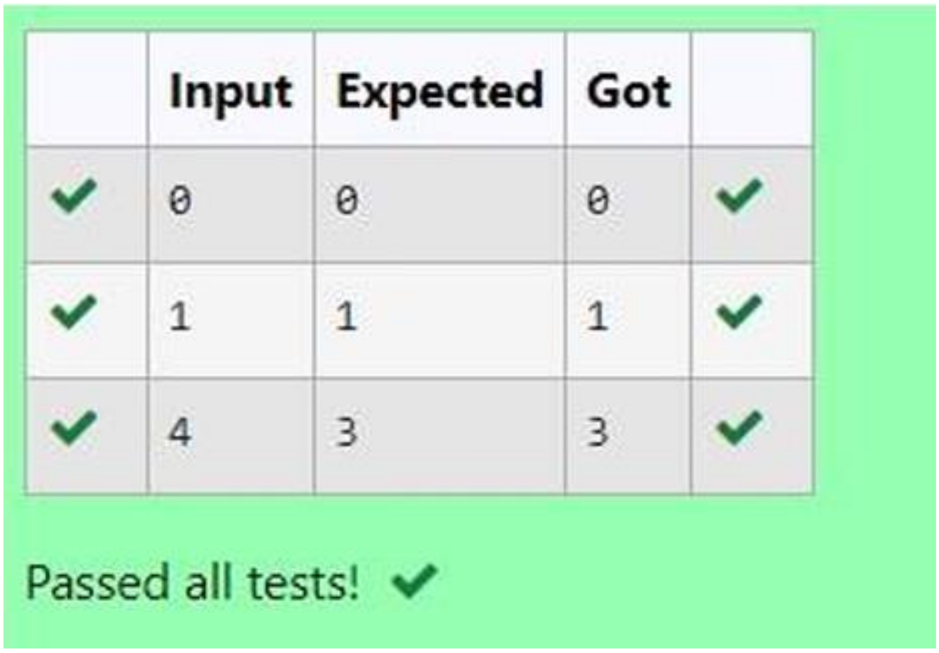
1. Initialize Variables
2. Input Value
3. If n is 0, the Nth term is f0 (which is 0).
4. If n is 1, the Nth term is f1 (which is 1).
5. Use a for loop to iterate from 2 to n.
6. In each iteration, calculate the next Fibonacci number as the sum of the previous two numbers (f2 = f0 + f1).
7. Update f0 and f1 to the next pair of Fibonacci numbers.
8. Output Result

PROGRAM:

```
#include <stdio.h>


int main() {

  int n, f0 = 0, f1 = 1, f2;

  scanf("%d", &n);


  if (n == 0) {

    printf("%d\n", f0);

  } else if (n == 1) {

    printf("%d\n",  f1);

  } else {

    for (int i = 2; i <= n; ++i) {

      f2 = f0 + f1;

      f0 = f1;

      f1 = f2;

    }
```

```
        printf("%d\n",  f2);
    }

    return 0;
}
```

OUTPUT:



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

Passed all tests! ✓

RESULT:

Thus, the program is executed successfully.

PROGRAM 13:

AIM:

Write a C program to find powers of integers.

ALGORITHM:

1: Initialize y, x and p as integers.

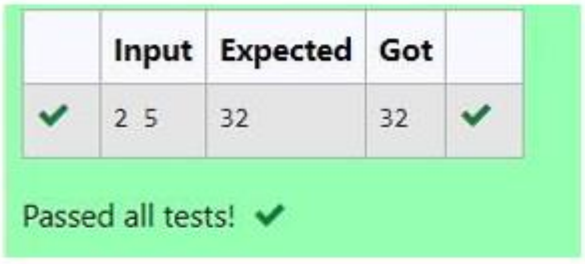
2: Take an input from the user for x and y.

3: calculate p as p=pow(x,y) and display p.

PROGRAM:

```
 #include <stdio.h>;

#include <Math.h>;

int main() {

            int y,x,p;

             scanf("%d%d",&x,&y);

             p=pow(x,y);

             printf("%d",p);

}
```

OUTPUT:



	Input	Expected	Got	
✓	2 5	32	32	✓

Passed all tests! ✓

RESULT:

Thus, the program is executed successfully.

PROGRAM 14:

AIM:

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

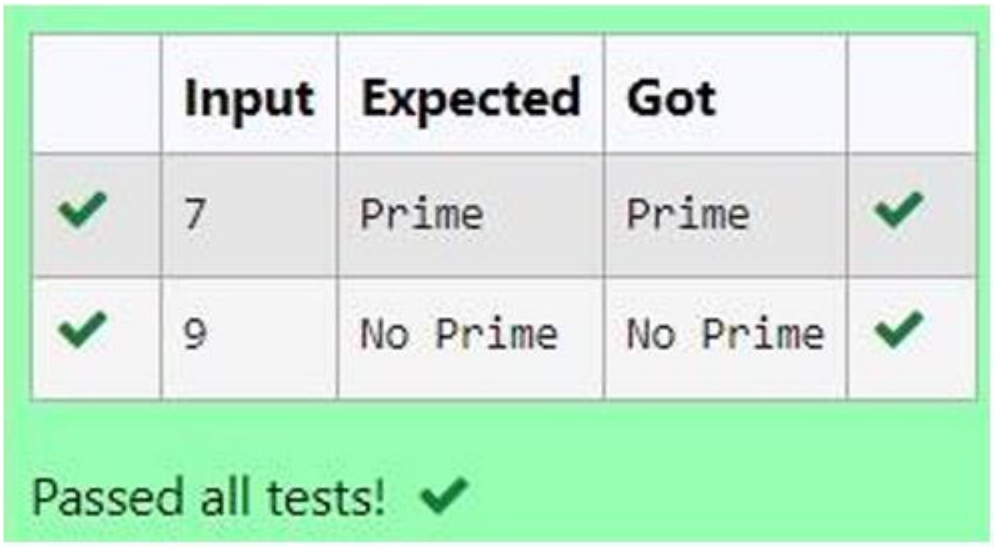
ALGORITHM:

1. Initialize Variable
2. Input Value
3. If n is less than or equal to 1, it is not a prime number.
4. For numbers greater than 1, check divisibility from 2 to the square root of n.
5. If n is divisible by any number in this range, it is not a prime number.
6. Otherwise, it is a prime number.
7. Output Result

PROGRAM:

```
#include <stdio.h>

#include <math.h>


int main() {

  int n, i, isPrime = 1;

  scanf("%d", &n);


  if (n <= 1) {

    isPrime = 0;

  } else {

    for (i = 2; i <= sqrt(n); i++) {

      if (n % i == 0) {

        isPrime = 0;

        break;

      }

    }

  }
  if (isPrime) {
    printf(”Prime ");
  } else {
    printf(“No Prime");
  }

  return 0;
}
```

OUTPUT:



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

Passed all tests! ✓

RESULT:

Thus, the program is executed successfully.

PROGRAM 15:

AIM:

Write a C program to find reverse of integer

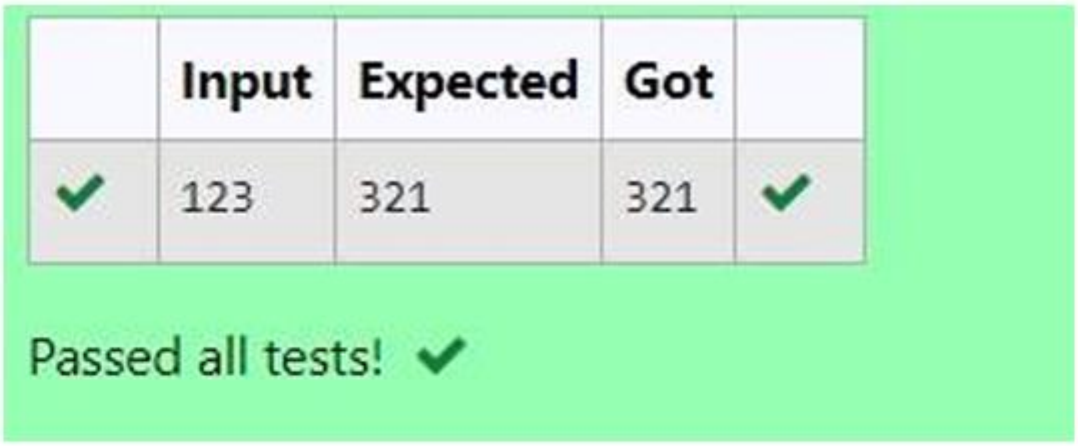
ALGORITHM:

1. Initialize Variables
2. Input Value
3. Use a while loop to iterate as long as n is not zero.
4. In each iteration, calculate the remainder of n divided by 10 (rem = n % 10).
5. Update the reversed number by multiplying rev by 10 and adding the remainder (rev = rev * 10 + rem).
6. Divide n by 10 to remove the last digit (n = n / 10).
7. Output Result

PROGRAM:

```
#include <stdio.h>


int main() {
  int n, rev = 0, rem;
  scanf("%d", &n);


  while (n != 0) {
    rem = n % 10;
    rev = rev * 10 + rem;
    n /= 10;
  }


  printf("%d\n", rev);
  return 0;
}
```

OUTPUT:



	Input	Expected	Got	
✓	123	321	321	✓

Passed all tests! ✓

RESULT:

Thus, the program is executed successfully.