

**RAJALAKSHMI ENGINEERING COLLEGE**

**RAJALAKSHMI NAGAR, THANDALAM – 602105**



**RAJALAKSHMI  
ENGINEERING COLLEGE**

**CS23331  
DESIGN AND ANALYSIS OF ALGORITHM**

**Laboratory Observation NoteBook**

**Name : N Subramanian**

**Year/Branch/Section : II/CSE/F**

**Register No. : 230701348**

**Semester : III**

**Academic Year: 2024-25**

# **01 - BASIC C PROGRAMMING**



**Ex. No. : 1.1**

**Date: 12.08.24**

**Register No.: 230701348**

**Name: N Subramanian**

---

### **AIM:**

Given two numbers, write a c program to swap two numbers.

For example:

INPUT	RESULT
10 20	20 10

### **ALGORITHM:**

Step 1: Start

Step 2: Get two numbers from the user, a and b.

Step 3: Store the value of a in a temporary variable called temp.

Step 4: Change the value of a to be the value of b. Set the value of b to the value stored in temp.

Step 5: Show the new values of a and b on the screen.

Step 6: End



## PROGRAM:

```
1 #include<stdio.h>
2 int main(){
3     int a,b;
4     scanf("%d %d",&a,&b);
5     int temp=a;
6     a=b;
7     b=temp;
8     printf("%d %d",a,b);
9 }
```

## OUTPUT:

	Input	Expected	Got	
✓	10 20	20 10	20 10	✓

Passed all tests! ✓

## RESULT:

Hence the above program has been executed successfully.

**Ex. No. : 1.2**

**Date: 12.08.24**

**Register No.: 230701348**

**Name: N Subramanian**

---

**AIM:**

Write a c program to check the eligibility of admission based on the following criteria:

Marks in Maths  $\geq 65$

Marks in Physics  $\geq 55$

Marks in Chemistry  $\geq 50$

or

Total in all three subjects  $\geq 180$

**SAMPLE TEST CASE:**

1. INPUT: 70 60 80

OUTPUT: The candidate is eligible

2. INPUT: 50 60 40

OUTPUT: The candidate is not eligible



## ALGORITHM:

Step 1: Start

Step 2: Read the marks for three subjects (a, b, c) from the user.

Step 3: Calculate the total marks,  $tot = a + b + c$ .

Step 4: Check if  $(a \geq 65 \text{ and } b \geq 55 \text{ and } c \geq 50)$  or  $tot \geq 180$ .

Step 5: If the condition is true, print "The candidate is eligible", otherwise print "The candidate is not eligible".

Step 6: End

## PROGRAM:

```
1  #include<stdio.h>
2  int main(){
3      int a,b,c;
4      scanf("%d %d %d",&a,&b,&c);
5      if((a>=65&&b>=55&&c>=50)|| (a+b+c)>=180){
6          printf("The candidate is eligible");
7      }else{
8          printf("The candidate is not eligible");
9      }
10 }
```

## 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! ✓

## RESULT:

Hence the above program has been executed successfully.

**Ex. No. : 1.3**

**Date: 12.08.24**

**Register No.: 230701348**

**Name: N Subramanian**

---

**AIM:**

Malini goes to BestSave supermarket to buy grocery.  
Supermarket provides 10% discount on bill amount B when  
the B is more than Rs.2000.

The program must find the final payable amount A.

**SAMPLE TEST CASE:**

1. INPUT: 1900

OUTPUT: 1900

2. INPUT: 3000

OUTPUT: 2700





## ALGORITHM:

Step 1: Start

Step 2: Read the value of a from the user.

Step 3: If  $a > 2000$ , calculate 10% of a and subtract from a store it in a ( $a = a - a * 0.1$ ), otherwise remains uninitialized.

Step 4: Print the RESULT a.

Step 5: End

## PROGRAM:

```
1  #include<stdio.h>
2  int main(){
3      int a;
4      scanf("%d",&a);
5      if(a>2000){
6          a=a-a*0.1;
7      }
8      printf("%d",a);
9  }
```

OUTPUT:

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

Passed all tests! ✓

RESULT:

Hence the above program has been executed successfully.

**Ex. No. : 1.4**

**Date: 12.08.24**

**Register No.: 230701348**

**Name: N Subramanian**

---

**AIM:**

Baba is very kind to beggars and everyday baba donates half of the amount he has when ever the beggar requests him. The money M left in Baba's hand is passed as the input and the numbers of beggars B is input. The program must print the money baba had in the beginning of the day.

**SAMPLE TEST CASE:**

1. INPUT: 100 2

2. OUTPUT: 400

**Explanation:**

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



## ALGORITHM:

Step 1: Start

Step 2: Read the value of a from the user.

Step 3: Read the value of b from the user.

Step 4: Calculate  $a * 2$  -- b times.

Step 5: Print the value of a.

Step 6: End

## PROGRAM:

```
1 #include<stdio.h>
2 int main(){
3     int a,b;
4     scanf("%d %d",&a,&b);
5     for(int i=0;i<b;i++){
6         a*=2;
7     }
8     printf("%d",a);
9 }
```

OUTPUT:

	Input	Expected	Got	
✓	100 2	400	400	✓

Passed all tests! ✓

RESULT:

Hence the above program has been executed successfully.

**Ex. No. : 1.5**

**Date: 12.08.24**

**Register No.: 230701348**

**Name: N Subramanian**

---

### **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.

### **SAMPLE TEST CASE:**

**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:

Step 1: Start

Step 2: Read the value of s from the user and initialize  $r = s$ .

Step 3: Read the value of n from the user.

Step 4: Loop from  $i = 0$  to  $n-2$ :

a. Add 200 to r ( $r = r + 200$ ).

b. Add the updated value of r to s ( $s = s + r$ ).

Step 5: Print the final value of s.

Step 6: End

## PROGRAM:

```
1  #include<stdio.h>
2  int main(){
3      int a,b,sum=0;
4      scanf("%d %d",&a,&b);
5      for(int i=0;i<b;i++){
6          sum+=a;
7          a+=200;
8      }
9      printf("%d",sum);
10 }
```

OUTPUT:

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

Passed all tests! ✓

RESULT:

Hence the above program has been executed successfully.



**Ex. No. : 1.6**

**Date: 12.08.24**

**Register No.: 230701348**

**Name: N Subramanian**

---

**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.

**SAMPLE TEST CASE:**

**1. INPUT: 2**

40

7

**OUTPUT: 35 28 21 14 7**



## ALGORITHM:

Step 1: Start

Step 2: Read the values of m, n, and x from the user.

Step 3: Loop through i from n to m (decreasing):

- a. If i is divisible by x ( $i \% x == 0$ ), print i.
- b. Otherwise, continue the loop.

Step 4: End

## PROGRAM:

```
1  #include<stdio.h>
2  int main(){
3      int a,b,c;
4      scanf("%d %d %d",&a,&b,&c);
5      for(int i=b;i>=a;i--){
6          if(i%c==0){
7              printf("%d ",i);
8          }
9      }
10 }
```

OUTPUT:

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

Passed all tests! ✓

RESULT:

Hence the above program has been executed successfully.

**Ex. No. : 1.7**

**Date: 12.08.24**

**Register No.: 230701348**

**Name: N Subramanian**

---

**AIM:**

Write a c program to find the quotient and reminder of given number.

For example:

INPUT	RESULT
12	4
3	0

**ALGORITHM:**

Step 1: Start

Step 2: Read the values of n and m from the user.

Step 3: Calculate the quotient by performing integer division  $n / m$  and print the RESULT.

Step 4: Calculate the remainder by performing  $n \% m$  and print the RESULT.

Step 5: End



## PROGRAM:

```
1 #include<stdio.h>
2 int main(){
3     int a,b;
4     scanf("%d %d",&a,&b);
5     int q=a/b,r=a%b;
6     printf("%d\n%d",q,r);
7 }
```

## OUTPUT:

	Input	Expected	Got	
✓	12	4	4	✓
	3	0	0	

Passed all tests! ✓

## RESULT:

Hence the above program has been executed successfully.

**Ex. No. : 1.8**

**Date: 12.08.24**

**Register No.: 230701348**

**Name: N Subramanian**

---

**AIM:**

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

For example:

INPUT	RESULT
10 20 30	30

**ALGORITHM:**

Step 1: Start

Step 2: Read the values of n1, n2, and n3 from the user.

Step 3: Check if n1 is greater than both n2 and n3. If true, print n1.

Step 4: Otherwise, check if n2 is greater than both n1 and n3. If true, print n2.

Step 5: If neither of the above conditions is true, print n3.

Step 6: End



## PROGRAM:

```
1 #include<stdio.h>
2 int main(){
3     int a,b,c;
4     scanf("%d %d %d",&a,&b,&c);
5     if(a>b){
6         if(a>c){
7             printf("%d",a);
8         }else{
9             printf("%d",c);
10        }
11    }else{
12        if(b>c){
13            printf("%d",b);
14        }else{
15            printf("%d",c);
16        }
17    }
18 }
```

## OUTPUT:

	Input	Expected	Got	
✓	10 20 30	30	30	✓

Passed all tests! ✓

## RESULT:

Hence the above program has been executed successfully.

**Ex. No. : 1.9**

**Date: 12.08.24**

**Register No.: 230701348**

**Name: N Subramanian**

---

**AIM:**

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

For example:

INPUT	RESULT
12	Even
11	Odd

**ALGORITHM:**

Step 1: Start

Step 2: Read the value of n from the user.

Step 3: Check if n is even by evaluating the condition  $n \% 2 == 0$ .

Step 4: If the condition is true, print "Even".

Step 5: If the condition is false, print "Odd".

Step 6: End





## PROGRAM:

```
1 #include<stdio.h>
2 int main(){
3     int a;
4     scanf("%d",&a);
5     if(a%2==0){
6         printf("Even");
7     }else{
8         printf("Odd");
9     }
10 }
```

## OUTPUT:

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

Passed all tests! ✓

**Correct**

Marks for this submission: 1.00/1.00.

## RESULT:

Hence the above program has been executed successfully.

**Ex. No. : 1.10**

**Date: 12.08.24**

**Register No.: 230701348**

**Name: N Subramanian**

---

**AIM:**

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

For example:

INPUT	RESULT
5	120

**ALGORITHM:**

Step 1: Start

Step 2: Read the value of n from the user.

Step 3: Initialize a variable f to 1.

Step 4: Loop through i from 2 to n:

    a. Multiply f by i ( $f = f * i$ ).

Step 5: Print the value of f, which is the factorial of n.

Step 6: End



## PROGRAM:

```
1  #include<stdio.h>
2  int fact(int n){
3      if(n==0){
4          return 1;
5      }
6      return n*fact(n-1);
7  }
8  int main(){
9      int a;
10     scanf("%d",&a);
11     printf("%d",fact(a));
12 }
```

## OUTPUT:

	Input	Expected	Got	
✓	5	120	120	✓

Passed all tests! ✓

**Correct**

Marks for this submission: 1.00/1.00.

## RESULT:

Hence the above program has been executed successfully.

**Ex. No. : 1.11**

**Date: 12.08.24**

**Register No.: 230701348**

**Name: N Subramanian**

---

**AIM:**

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

For example:

INPUT	RESULT
3	6

**ALGORITHM:**

Step 1: Start

Step 2: Read the value of n from the user.

Step 3: Initialize a variable s to 0.

Step 4: Loop through i from 1 to n:

    a. Add i to s ( $s = s + i$ ).

Step 5: Print the value of s, which is the sum of the first n natural numbers.

Step 6: End



## PROGRAM:

```
1 #include<stdio.h>
2 int main(){
3     int a,sum=0;
4     scanf("%d",&a);
5     for(int i=1;i<=a;i++){
6         sum+=i;
7     }
8     printf("%d",sum);
9 }
```

## OUTPUT:

	Input	Expected	Got	
✓	3	6	6	✓

Passed all tests! ✓

Correct

## RESULT:

Hence the above program has been executed successfully.

**Ex. No. : 1.12**

**Date: 12.08.24**

**Register No.: 230701348**

**Name: N Subramanian**

---

**AIM:**

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

For example:

INPUT	RESULT
0	0
1	1
4	3

**ALGORITHM:**

Step 1: Start

Step 2: Read the value of n from the user.

Step 3: Initialize  $f0 = 0$ ,  $f1 = 1$ , and  $f2$ .

Step 4: If n is 0, print  $f0$ ; if n is 1, print  $f1$ .

Step 5: For i from 1 to  $n - 1$ , calculate  $f2 = f1 + f0$ , then update  $f0$  to  $f1$  and  $f1$  to  $f2$ .

Step 6: Print the value of  $f2$ . End.



## PROGRAM:

```
1  #include<stdio.h>
2  int main(){
3      int a=0,b=1,n;
4      scanf("%d",&n);
5      while(n--){
6          int c=a+b;
7          a=b;
8          b=c;
9      }
10     printf("%d",a);
11 }
```

## OUTPUT:

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

Passed all tests! ✓

## RESULT:

Hence the above program has been executed successfully.

**Ex. No. : 1.13**

**Date: 12.08.24**

**Register No.: 230701348**

**Name: N Subramanian**

---

**AIM:**

Write a C program to find the power of integers.

For example:

INPUT	RESULT
2 5	32

**ALGORITHM:**

Step 1: Start

Step 2: Read the value of n from the user and store it in t.

Step 3: Read the value of m from the user.

Step 4: For i from 1 to m - 1, multiply n by t.

Step 5: Print the final value of n.

Step 6: End





## PROGRAM:

```
1 #include<stdio.h>
2 int main(){
3     int a,b,e=1;
4     scanf("%d %d",&a,&b);
5     for(int i=0;i<b;i++){
6         e*=a;
7     }
8     printf("%d",e);
9 }
```

## OUTPUT:

	Input	Expected	Got	
✓	2 5	32	32	✓

Passed all tests! ✓

## RESULT:

Hence the above program has been executed successfully.

**Ex. No. : 1.14**

**Date: 12.08.24**

**Register No.: 230701348**

**Name: N Subramanian**

---

**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:**

Step 1: Start

Step 2: Read the value of n from the user.

Step 3: Initialize a variable flag to 0.

Step 4: For i from 2 to n - 1, check if  $n \% i == 0$ . If true, set flag to 1 and break the loop.

Step 5: If flag is 0, print "Prime"; otherwise, print "No Prime".

Step 6: End



## PROGRAM:

```
1 #include<stdio.h>
2 int main(){
3     int a;
4     scanf("%d",&a);
5     if(a==1){
6         printf("No Prime");
7     }else if(a==2){
8         printf("Prime");
9     }else{
10        int flag=0;
11        for(int i=2;i<=a/2;i++){
12            if(a%i==0){
13                flag=1;
14                break;
15            }
16        }
17        if(flag){
18            printf("No Prime");
19        }else{
20            printf("Prime");
21        }
22    }
23 }
```

## OUTPUT:

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

Passed all tests! ✓

## RESULT:

Hence the above program has been executed successfully.

**Ex. No. : 1.15**

**Date: 12.08.24**

**Register No.: 230701348**

**Name: N Subramanian**

---

**AIM:**

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

**ALGORITHM:**

Step 1: Start

Step 2: Read the value of n from the user.

Step 3: Initialize a variable d to 0.

Step 4: While n is not 0, do the following:

- a. Set r to  $n \% 10$ .
- b. Update d as  $d = (d * 10) + r$ .
- c. Update n as  $n = n / 10$ .

Step 5: Print the value of d, which is the reversed number.

Step 6: End



## PROGRAM:

```
1 #include<stdio.h>
2 int main(){
3     int a,rev=0,r;
4     scanf("%d",&a);
5     while(a>0){
6         r=a%10;
7         a/=10;
8         rev=rev*10+r;
9     }
10    printf("%d",rev);
11 }
```

## OUTPUT:

	Input	Expected	Got	
✓	123	321	321	✓

Passed all tests! ✓

## RESULT:

Hence the above program has been executed successfully.