RAJALAKSHMI ENGINEERING COLLEGE RAJALAKSHMI NAGAR, THANDALAM – 602 105

CS23331 - Design and Analysis of Algorithms

Laboratory Record Notebook

NAME : HARISH R P
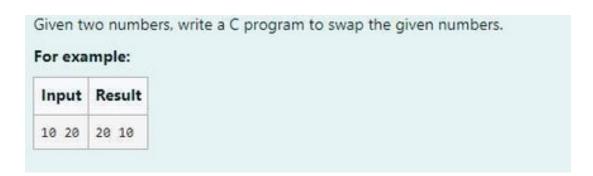
REG.NO : 230701106

DEPT : B E COMPUTER SCIENCE AND ENGINEERING – B

Week–1 Basic C Programming

QUESTION

AIM :

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

For example:

Input	Result
10 20	20 10

ALGORITHM :

Step 1: Start

Step 2: Input integers x and y

Step 3: Store the value of x in temp Step 4: Assign the value of y to x Step 5: Assign the value of temp to y Step 6: Print x and y

Step 7: Stop

PROGRAM :

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

}
```

OUTPUT:

	Input	Expected	Got	
✔	10 20	20 10	20 10	✔

Passed all tests! ✔

RESULT :

The above program is executed successfully.

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:

Step 1: Start

Step 2: Input marks for Physics (p), Chemistry (c), and Math (m)

Step 3: Check if m >= 65, p >= 55, c >= 50 or if the total marks m + p + c >= 180

Step 4: If true, print "The candidate is eligible"; else, print "The candidate is not eligible" Step 5: Stop

PROGRAM :

```c
#include<stdio.h>
int main()
{
    int m,p,c,t;
    scanf("%d %d %d",&m,&p,&c);
    t=m+p+c;
    if(m>=65 && p>=55 && c>=50){
        printf("The candidate is eligible");
    }
    else if(t>=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! ✔

RESULT:

The above program is executed successfully.

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:

Step 1: Start

Step 2: Input the bill amount b

Step 3: If b > 2000, calculate a discount of 10% and subtract it from b to get the final amount f Step 4: If b <= 2000, set f = b

Step 5: Print f Step 6: Stop

PROGRAM:

```c
#include<stdio.h>
int main()
{
    int b;
    scanf("%d",&b);
    if(b>2000)
    {
        int p=(0.1*b);
        int pay=(b-p);
        printf("%d",pay);
    }
    else{
        printf("%d",b);
    }
}
```

OUTPUT:

	Input	Expected	Got	
✔	1900	1900	1900	✔
✔	3000	2700	2700	✔

Passed all tests! ✔

RESULT:

The above program is executed successfully.

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:

Step 1: Start

Step 2: Input integers m and b

Step 3: While b is not zero, double the value of m and decrement b by 1 Step 4: Print the value of m

Step 5: Stop

PROGRAM:

```c
#include<stdio.h>
int main(){
    int m,b;
    scanf("%d %d",&m,&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! ✔

RESULT:

The above program is executed successfully.

AIM :

ALGORITHM:

Step 1: Start

Step 2: Input integers i and d Step 3:

Initialize s with the value of i

Step 4: While d > 1, add 200 to i, add i to s, and decrement d by 1 Step 5: Print the value of s

Step 6: Stop

PROGRAM:

```c
#include<stdio.h>
int main(){
    int i,n,a=0,t=0;
    scanf("%d %d",&i,&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! ✔

RESULT:

The above program is executed successfully.

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:

Step 1: Start

Step 2: Input integers m, n, and x Step 3: Initialize i with the value of n

Step 4: While i >= m, if i is divisible by x, print i Step 5: Decrement i by 1

Step 6: Stop

PROGRAM:

```c
#include<stdio.h>
int main()
{
    int n,m,x;
    scanf("%d %d %d",&n,&m,&x);
    while(m>=n){
        if (m%x==0){
            printf("%d ", m);
        }
        m--;
    }
}
```

OUTPUT:

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

Passed all tests! ✔

RESULT:

The above program is executed successfully.

AIM:

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

For example:

Input	Result
12	4
3	0

ALGORITHM:

Step 1: Start

Step 2: Input integers a and d

Step 3: Calculate the quotient q = a / d and remainder r = a % d Step 4: Print q and r

Step 5: Stop

PROGRAM:

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

OUTPUT:

	Input	Expected	Got	
✔	12 3	4 0	4 0	✔

Passed all tests! ✔

RESULT:

The above program is executed successfully.

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: Input three integers a, b, and c

Step 3: Check which of the three integers is the largest Step 4: Print the largest integer

Step 5: Stop

PROGRAM:

```c
#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);
    }
}
```

OUTPUT:

	Input	Expected	Got	
✔	10 20 30	30	30	✔

Passed all tests! ✔

RESULT:

The above program is executed successfully.

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

For example:

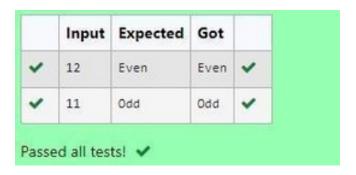| Input	Result
12	Even
11	Odd

AIM:

ALGORITHM:

Step 1: Start

Step 2: Input an integer a

Step 3: Check if a is even or odd

Step 4: Print "Even" if a is even; otherwise, print "Odd" Step 5: Stop

PROGRAM:

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

	Input	Expected	Got	
✔	12	Even	Even	✔
✔	11	Odd	Odd	✔

Passed all tests! ✔

OUTPUT:

RESULT:

The above program is executed successfully.

AIM:

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

For example:

Input	Result
5	120

ALGORITHM:

Step 1: Start

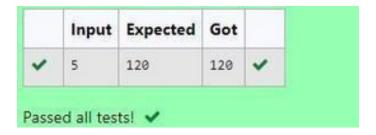Step 2: Input an integer a Step 3: Set x = a

Step 4: While x > 1, decrement x by 1 and multiply it with a Step 5: Print the final value of a

Step 6: Stop

PROGRAM:

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

OUTPUT:

	Input	Expected	Got	
✔	5	120	120	✔

Passed all tests! ✔

RESULT:

The **above** program is executed successfully.

AIM:

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

For example:

Input	Result
3	6

ALGORITHM:

Step 1: Start

Step 2: Input an integer a Step 3: Initialize $b = 0$

Step 4: While $a \mathrel{!=} 0$, add a to b and decrement a by 1 Step 5: Print the value of b

Step 6: Stop

PROGRAM:

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

:

OUTPUT:

	Input	Expected	Got	
✔	3	6	6	✔

Passed all tests! ✔

RESULT:

The above program is executed successfully.

AIM:

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

For example:

Input	Result
0	0
1	1

ALGORITHM:

Step 1: Start

Step 2: Input an integer n

Step 3: Define a recursive function fib(n) that returns fib(n–1) + fib(n–2) for n > 1 and n for n <= 1 Step 4: Print the result of fib(n)

Step 5: Stop

PROGRAM:

```c
#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;
}
```

OUTPUT:

	Input	Expected	Got	
✔	0	0	0	✔
✔	1	1	1	✔
✔	4	3	3	✔

RESULT:

The above program is executed successfully.

AIM:

Write a C program to find the power of integers.

input:

a b

output:

a^b value

ALGORITHM:

Step 1: Start

Step 2: Input integers a and b Step 3: Initialize i = 0 and p = 1

Step 4: While i < b, multiply p with a and increment i by 1 Step 5: Print the value of p

Step 6: Stop

PROGRAM:

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

OUTPUT:

	Input	Expected	Got	
✔	2 5	32	32	✔

RESULT:

The above program is executed successfully.

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: Input an integer n

Step 3: For each number i from 2 to n−1, check if n % i == 0 Step 4: If divisible, set flag = 1 and break; else, set flag = 0

Step 5: If flag == 0, print "Prime"; else, print "No Prime" Step 6: Stop

PROGRAM:

```c
#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");
    }
}
```

OUTPUT:

RESULT:

The above program is executed successfully.

AIM:

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

ALGORITHM:

Step 1: Start
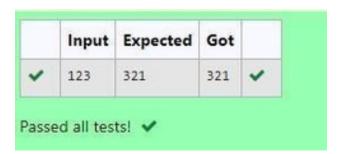
Step 2: Input an integer n Step 3: Initialize rev = 0

Step 4: While n != 0, calculate the remainder rem = n % 10 Step 5: Update rev = rev * 10 + rem and divide n by 10 Step 6: Print rev

Step 7: Stop

PROGRAM:

```c
#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);
}
```

OUTPUT:

	Input	Expected	Got	
✔	123	321	321	✔

Passed all tests! ✔

RESULT:

The above program is executed successfully.

Name : GANESHAN M**Register No :** 230701514**Branch :** B.E COMPUTER SCIENCE   AND ENGINEERING**Year : II****Section : A****Semester : III****Academic Year:** 2024-25