RAJALAKSHMI ENGINEERING COLLEGE

RAJALAKSHMI NAGAR, THANDALAM – 602 105

CS23331 **DESIGN AND ANALYSIS OF ALGORITHM**
Laboratory Observation NoteBook

Name : S Vishwak
Year/Branch/Section : II/CSE/F
Register No. : 230701385
Semester : III
Academic Year: 2024-25

01 – BASIC C PROGRAMMING

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:

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

OUTPUT:

	Input	Expected	Got	
✔	10 20	20 10	20 10	✔

Passed all tests! ✔

Correct
Marks for this submission: 1.00/1.00.

RESULT:

Hence the above program has been executed successfully.

AIM:

Write a c program to check the eligibility of admission 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 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 (m, p, c) from the user.

Step 3: Calculate the total marks, tot = m + p + c.

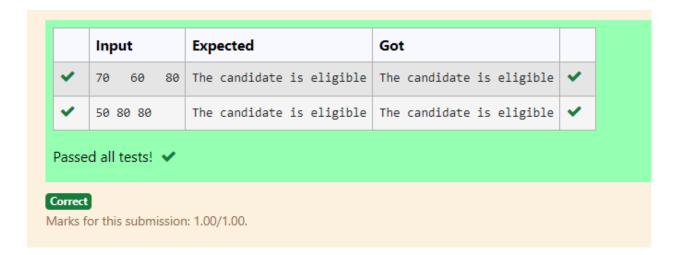Step 4: Check if (m >= 65 and p >= 55 and c >= 50) or tot >= 180.

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

Step 6: End

PROGRAM:

```c
#include<stdio.h>
int main()
{
    int m,p,c,ans=0;
    scanf("%d%d%d",&m,&p,&c);
    ans=m+p+c;;
    if((m>=65 && p>=55 && c>=50)||ans>=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:

Hence the above program has been executed successfully.

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 b from the user.

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

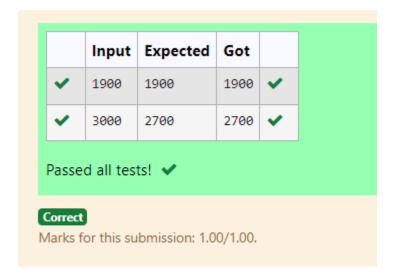Step 4: Subtract a from b to calculate the final amount.

Step 5: Print the RESULT (b - a).

Step 6: End

PROGRAM:

```
1  #include<stdio.h>
2  int main()
3  {
4      int b;
5      scanf("%d",&b);
6      if(b>2000)
7      {
8        b=b-(b*0.1);
9      }
10     printf("%d",b);
11 }
```

OUTPUT:

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

Passed all tests! ✔

Correct
Marks for this submission: 1.00/1.00.

RESULT:

Hence the above program has been executed successfully.

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

   OUTPUT: 400

Explaination:

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

ALGORITHM:

Step 1: Start

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

Step 3: Read the value of b from the user.
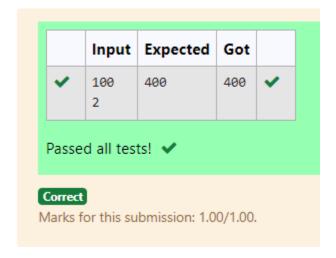
Step 4: Calculate t = 2 * m * b.

Step 5: Print the value of t.

Step 6: End

PROGRAM:

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

OUTPUT:

	Input	Expected	Got	
✔	100 2	400	400	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

RESULT:

  Hence the above program has been executed successfully.

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:

1. 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).
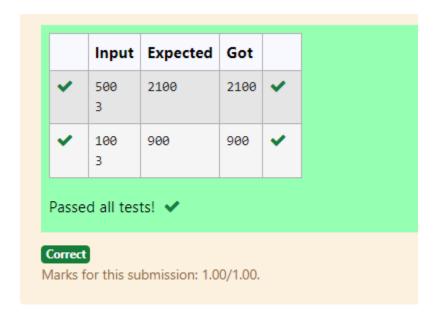
    b. Add the updated value of r to s (s = s + r).

Step 5: Print the final value of s.

Step 6: End

PROGRAM:

```
1   #include<stdio.h>
2   int main()
3 ▾ {
4       int i,n,pi=0;
5       scanf("%d%d",&i,&n);
6 ▾    while(n>0){
7           pi=pi+i;
8           i=i+200;
9           n--;
10      }
11      printf("%d",pi);
12  }
```

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:

Hence the above program has been 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.

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.
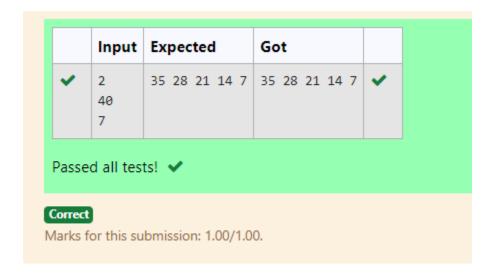
    b. Otherwise, continue the loop.

Step 4: End

PROGRAM:

```c
#include<stdio.h>
int main()
{
    int m,n,x;
    scanf("%d%d%d",&m,&n,&x);
    for(int i=n;i>=m;i--)
    {
        if(i%x==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:

Hence the above program has been executed successfully.

AIM:

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

For example:

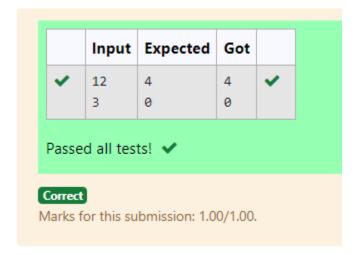| INPUT	RESULT
12	4
3	0

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 ▾ {
4      int a,b,quo=0,rem=0;
5      scanf("%d%d",&a,&b);
6      quo=a/b;
7      rem=a%b;
8      printf("%d\n%d",quo,rem);
9  }
```

OUTPUT:

	Input	Expected	Got	
✔	12	4	4	✔
	3	0	0	

Passed all tests! ✔

Correct
Marks for this submission: 1.00/1.00.

RESULT:

Hence the above program has been 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

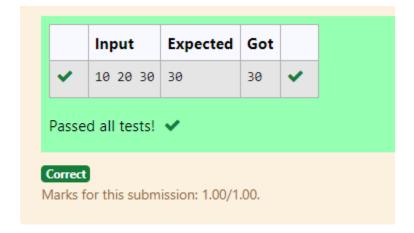Step 2: Read the values of n1, n2, and n3 from the user.

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 ▾ {
4       int a,b,c;
5       scanf("%d%d%d",&a,&b,&c);
6 ▾    if(a>b && a>c){
7           printf("%d",a);
8       }
9 ▾    else if(b>a && b>c){
10          printf("%d",b);
11      }
12 ▾   else{
13          printf("%d",c);
14      }
15 }
```

OUTPUT:

	Input	Expected	Got	
✔	10 20 30	30	30	✔

Passed all tests! ✔

Correct
Marks for this submission: 1.00/1.00.

RESULT:

Hence the above program has been executed successfully.

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:

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

Hence the above program has been 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: 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:

```c
#include<stdio.h>
int main()
{
    int n,fact=1;
    scanf("%d",&n);
    for(int i=1;i<=n;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:

Hence the above program has been executed successfully.

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:

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

OUTPUT:

	Input	Expected	Got	
✔	3	6	6	✔

Passed all tests! ✔

Correct
Marks for this submission: 1.00/1.00.

RESULT:

Hence the above program has been executed successfully.

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.

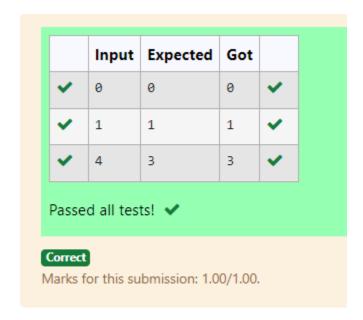Step 3: Initialize f0 = 0, f1 = 1, and f2.

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:

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

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:

Hence the above program has been executed successfully.

AIM:

Write a C program to find the power of integers.

For example:

INPUT	RESULT
2 5	32

ALGORITHM:

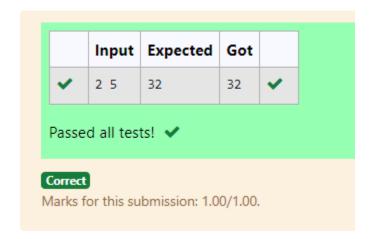Step 1: Start

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:

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

OUTPUT:

	Input	Expected	Got	
✔	2 5	32	32	✔

Passed all tests! ✔

Correct
Marks for this submission: 1.00/1.00.

RESULT:

Hence the above program has been 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

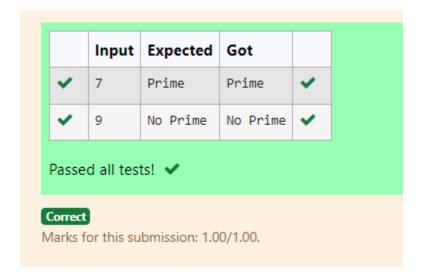Step 2: Read the value of n from the user.

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:

```c
1  #include<stdio.h>
2  int main()
3  {
4      int n,flag=1;
5      scanf("%d",&n);
6      for(int i=2;i<n/2;i++)
7      {
8          if(n%i==0){
9              flag=0;
10             break;
11         }
12         else{
13             flag=1;
14         }
15     }
16     if(flag==1)
17     {
18         printf("Prime");
19     }
20     else
21     {
22         printf("No Prime");
23     }
24 }
```

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:

Hence the above program has been executed successfully.

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.

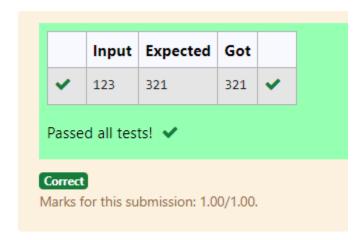Step 4: While n is not 0, do the following:

    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 ▾{
4      int n,a=0;
5      scanf("%d",&n);
6      while(n>0)
7 ▾   {
8          int rem=n%10;
9          a=(a*10)+rem;
10         n=n/10;
11     }
12     printf("%d",a);
13
14 }
```

OUTPUT:

	Input	Expected	Got	
✔	123	321	321	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

RESULT:

Hence the above program has been executed successfully.