

L PRACTICE LAB ASSIGNMENT 6

1. Define a function called hypotenuse that calculates the length of the hypotenuse of a right triangle when the other two sides are given. The function should take two arguments of type double and return the hypotenuse as a double.

CODE

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

```
#include <math.h>
```

```
double hypotenuse(double, double);
```

```
double main()
```

```
{
```

```
double A, B;
```

```
printf("\nEnter two values for the two sides of a right-angled triangle to find its  
HYPOTENUSE");
```

```
printf("\nEnter A: ");
```

```
scanf("%lf", &A);
```

```
printf("\nEnter B: ");
```

```
scanf("%lf", &B);
```

```
printf("\nThe HYPOTENUSE is %lf", hypotenuse(A, B));
```

```
}
```

```
double hypotenuse(double A, double B)
```

```
{
```

```
double n, C;
```

```
C = sqrt((A * A) + (B * B));
```

```
return C;
```

```
}
```

SS of the OUTPUT

```
#include <stdio.h>
#include <math.h>
double hypotenuse(double, double);
double main()
{
    double A, B;
    printf("\nEnter two values for the two sides of a right-angled triangle to find its HYPOTENUSE");
    printf("\nEnter A: ");
    scanf("%lf", &A);
    printf("\nEnter B: ");
    scanf("%lf", &B);
    printf("\nThe HYPOTENUSE is %lf", hypotenuse(A, B));
}
double hypotenuse(double A, double B)
{
    double n, C;
    C = sqrt((A * A) + (B * B));
    return C;
}
```

```
student@HP-280-G3-MT: ~/Desktop
student@HP-280-G3-MT:~$ pwd
/home/student
student@HP-280-G3-MT:~$ cd Desktop
student@HP-280-G3-MT:~/Desktop$ ls
a.out  me  q1.c  Untitled Folder
student@HP-280-G3-MT:~/Desktop$ gcc q1.c -lm
student@HP-280-G3-MT:~/Desktop$ ./a.out

Enter two values for the two sides of a right-angled triangle to find its HYPOTE
NUSE
Enter A: 3

Enter B: 4

The HYPOTENUSE is 5.000000student@HP-280-G3-MT:~/Desktop$
```

2. An integer number is said to be a perfect number if its factors, including 1 (but not the number itself), sum to the number. For example, 6 is a perfect number because $6 = 1 + 2 + 3$. Write a function `perfect()` that determines if parameter number is a perfect number. Use this function in a program that determines and prints all the perfect numbers between 1 and 1000. Print the factors of each perfect number to confirm that the number is indeed perfect.

CODE

```
#include <stdio.h>

int perfect(int);

int main()
{
    int n;

    printf("\nThe list of perfect numbers between 1 and 1000 are:");

    perfect(n);

    return 0;
}

int perfect(int n)
{
    int i, sum;

    for(n = 1; n <= 1000; n++)
    {
        sum = 0;

        for(i = 1; i < n; i++)
        {
            if(n%i == 0)
            {
                sum += i;
            }
        }

        if(sum == n)
```

```

{
    printf("\n%d is a PERFECT number", n);
    printf("\nIt's factors are: ");
    {
        for(i = 1; i < n; i++)
        {
            if(n%i == 0)
            {
                printf("%d ", i);
            }
        }
        printf("\n");
    }
}
}
}

```

SS of the OUTPUT

```

9  #include <stdio.h>
10 int perfect(int);
11 int main()
12 {
13     int n;
14     printf("\nThe list of perfect numbers between 1 and 1000 are:");
15     perfect(n);
16     return 0;
17 }
18 int perfect(int n)
19 {
20     int i, sum;
21     for(n = 1; n <= 1000; n++)
22     {
23         sum = 0;
24         for(i = 1; i < n; i++)
25         {
26             if(n%i == 0)
27             {
28                 sum += i;
29             }
30         }
31         if(sum == n)
32         {
33             printf("\n%d is a PERFECT number", n);
34             printf("\nIt's factors are: ");
35             {
36                 for(i = 1; i < n; i++)
37                 {
38                     if(n%i == 0)
39                     {
40                         printf("%d ", i);
41                     }
42                 }
43                 printf("\n");
44             }
45         }
46     }
47 }

```

The list of perfect numbers between 1 and 1000 are:
6 is a PERFECT number
It's factors are: 1 2 3
28 is a PERFECT number
It's factors are: 1 2 4 7 14
496 is a PERFECT number
It's factors are: 1 2 4 8 16 31 62 124 248
...Program finished with exit code 0
Press ENTER to exit console.

3. Write a program using function that takes an integer value and returns the number with its digits reversed. For example, given the number 7631, the function should return 1367.

CODE

```
#include <stdio.h>

int reverse(int);

int main()
{
    int n;

    printf("\nEnter any integer to find out its reverse: ");

    scanf("%d", &n);

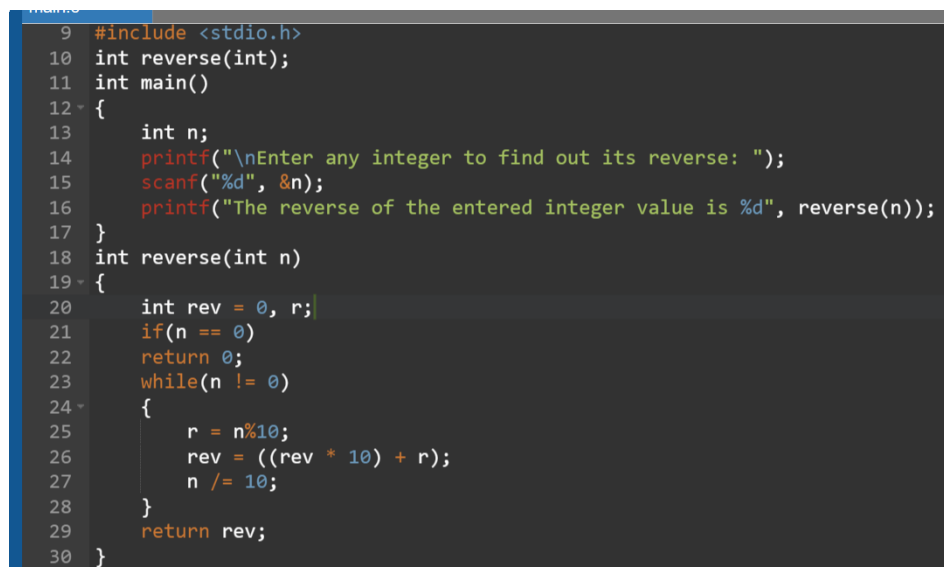
    printf("The reverse of the entered integer value is %d", reverse(n));
}

int reverse(int n)
{
    int rev = 0, r;

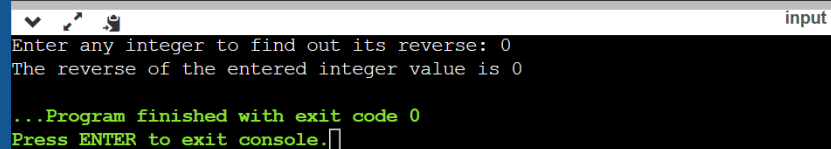
    if(n == 0)
        return 0;

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

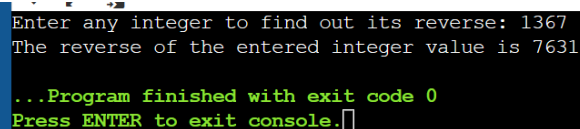
    return rev;
}
```



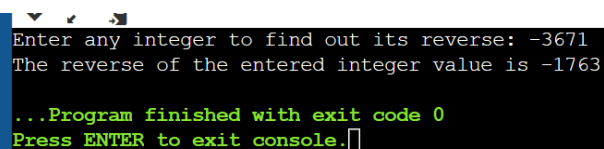
```
9  #include <stdio.h>
10 int reverse(int);
11 int main()
12 {
13     int n;
14     printf("\nEnter any integer to find out its reverse: ");
15     scanf("%d", &n);
16     printf("The reverse of the entered integer value is %d", reverse(n));
17 }
18 int reverse(int n)
19 {
20     int rev = 0, r;
21     if(n == 0)
22         return 0;
23     while(n != 0)
24     {
25         r = n%10;
26         rev = ((rev * 10) + r);
27         n /= 10;
28     }
29     return rev;
30 }
```



```
input
Enter any integer to find out its reverse: 0
The reverse of the entered integer value is 0
...Program finished with exit code 0
Press ENTER to exit console.
```



```
Enter any integer to find out its reverse: 1367
The reverse of the entered integer value is 7631
...Program finished with exit code 0
Press ENTER to exit console.
```



```
Enter any integer to find out its reverse: -3671
The reverse of the entered integer value is -1763
...Program finished with exit code 0
Press ENTER to exit console.
```

SS of the OUTPUT

4. Write a Program to print the Fibonacci Series with the help of functions. A Fibonacci Series is a series of numbers where the next number in the series is equal to the sum of previous 2 numbers. Example:-A Fibonacci series of 8 terms will be -0, 1, 1, 2, 3, 5, 8, 13.

CODE

```

#include<stdio.h>

int fibonacci(int, int, int, int);

int main()
{
    int t, a = 0, b = 1, n = a + b;

    printf("\nEnter the number of terms of the Fibonacci series you need: ");
    scanf("%d", &t);

    printf("A Fibonacci series of %d terms will be: 0, 1, ", t);

    fibonacci(t, a, b, n);

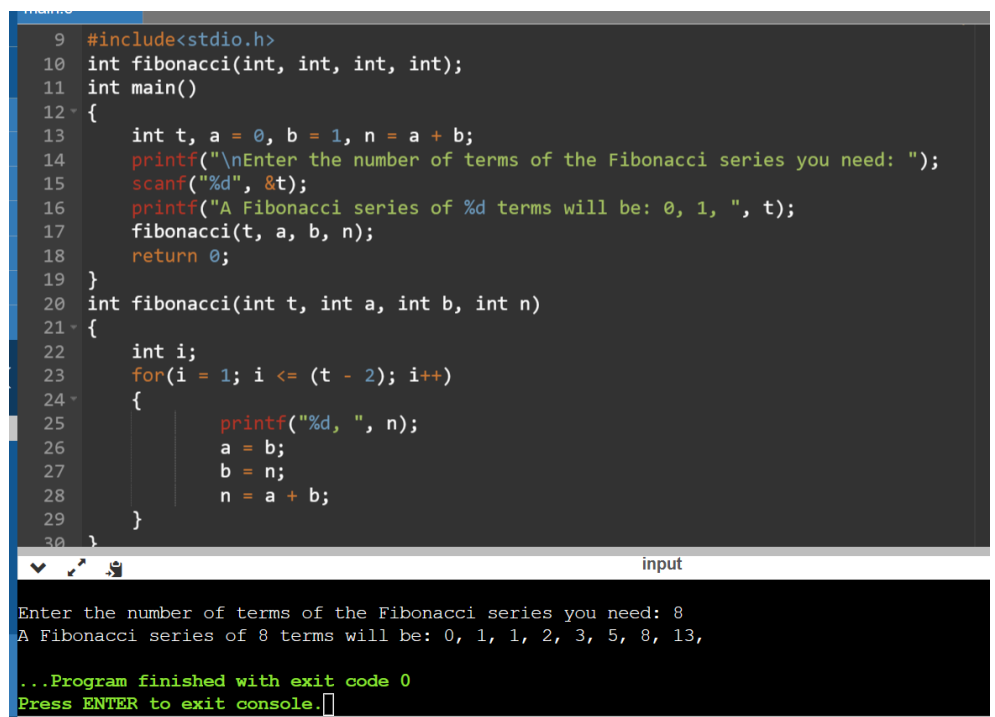
    return 0;
}

int fibonacci(int t, int a, int b, int n)
{
    int i;
    for(i = 1; i <= (t - 2); i++)
    {
        printf("%d, ", n);

        a = b;
        b = n;
        n = a + b;
    }
}

```

SS of the OUTPUT



```

9  #include<stdio.h>
10 int fibonacci(int, int, int, int);
11 int main()
12 {
13     int t, a = 0, b = 1, n = a + b;
14     printf("\nEnter the number of terms of the Fibonacci series you need: ");
15     scanf("%d", &t);
16     printf("A Fibonacci series of %d terms will be: 0, 1, ", t);
17     fibonacci(t, a, b, n);
18     return 0;
19 }
20 int fibonacci(int t, int a, int b, int n)
21 {
22     int i;
23     for(i = 1; i <= (t - 2); i++)
24     {
25         printf("%d, ", n);
26         a = b;
27         b = n;
28         n = a + b;
29     }
30 }

```

input

Enter the number of terms of the Fibonacci series you need: 8

A Fibonacci series of 8 terms will be: 0, 1, 1, 2, 3, 5, 8, 13,

...Program finished with exit code 0

Press ENTER to exit console.

5. Write a function multiple that determines for a pair of integers whether the second integer is a multiple of the first. The function should take two integer arguments and return 1(true) if the second is a multiple of the first and 0 (false) otherwise.

CODE

```

#include<stdio.h>

```

```

int multiple(int, int);

int main()
{
    int n, m;

    printf("\nEnter any integer: ");
    scanf("%d", &n);

    printf("\nEnter an integer to check if it's a multiple of the previous number: ");
    scanf("%d", &m);
    printf("%d", multiple(n, m));

    return 0;
}

int multiple(int n, int m)
{
    if(m%n == 0)
        return 1;
    else
        return 0;
}

```

SS of the OUTPUT

```

9  #include<stdio.h>
10 int multiple(int, int);
11 int main()
12 {
13     int n, m;
14     printf("\nEnter any integer: ");
15     scanf("%d", &n);
16     printf("\nEnter an integer to check if it's a multiple of the previous number: ");
17     scanf("%d", &m);
18     printf("%d", multiple(n, m));
19     return 0;
20 }
21 int multiple(int n, int m)
22 {
23     if(m%n == 0)
24         return 1;
25     else
26         return 0;
27 }

```

input

```

Enter any integer: 7

Enter an integer to check if it's a multiple of the previous number: 48
0

...Program finished with exit code 0
Press ENTER to exit console.

```

```

Enter any integer: 7

Enter an integer to check if it's a multiple of the previous number: 49
1

...Program finished with exit code 0
Press ENTER to exit console.

```