

FOUNDATION UNIVERSITY ISLAMABAD

SCHOOL OF SCIENCE AND TECHNOLOGY

LAB MANUAL 7

NAME: ALISHBA HAFEEZ

SECTION: 25(B)

SUBJECT: PF(programming fundamentals)

ROLL NO: 056

FORM NO: 1025234338

DEPARTMENT: BS IET

INSTRUCTOR: Mr. SHAMSHAIR

TASK 7.1:

Functions with No Arguments/One Argument

Write a C++ program which takes an integer ‘ N’ from user and calls these two functions to generate tables of all numbers between 1 to N. Your program should have the following interface.

CODE:

```
#include <iostream>
using namespace std;
int getNumber()
{
    int n;
    cout << "Enter value of N: ";
    cin >> n;
    return n;
}
void printTable(int num)
{
    cout << "\nTable of " << num << endl;
for (int i = 1; i <= 10; i++)
{
    cout << num << " x " << i << " = " << num * i << endl;
```

```

        }

    }

int main()
{
    int N = getNumber();

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

    return 0;
}

```

OUTPUT:

```

Enter value of N: 2

Table of 1
1 x 1 = 1
1 x 2 = 2
1 x 3 = 3
1 x 4 = 4
1 x 5 = 5
1 x 6 = 6
1 x 7 = 7
1 x 8 = 8
1 x 9 = 9
1 x 10 = 10

Table of 2
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
2 x 6 = 12
2 x 7 = 14
2 x 8 = 16
2 x 9 = 18
2 x 10 = 20

-----
Process exited after 8.318 seconds with return value 0
Press any key to continue . . .

```

TASK 7.2:

Functions with a Return Value

An integer is said to be a Perfect Number if the sum of its divisors, including 1 (but not itself), is equal to that number. For example, 6 is a perfect number, because $6 = 1+2+3$. Write a C++ function called ‘IsPerfect’ that takes an integer as parameter and returns 1 if the passed integer is a perfect number and 0 if it is not a perfect number. Use this function in a C++ program that reads in an integer value from user into a variable named ‘N’ and prints all perfect numbers between 1 and N. Print the

divisors of each perfect number to confirm that the number is indeed a perfect number. Your program should have the following interface.

CODE:

```
#include <iostream>
using namespace std;
int IsPerfect(int num)
{
    int sum = 0;

    for (int i = 1; i <= num / 2; i++)
    {
        if (num % i == 0)
        {
            sum = sum + i;
        }
    }

    if (sum == num)
        return 1;

    else
        return 0;
}

int main()
{
    int N;

    cout << "Enter a number: ";
    cin >> N;

    for (int i = 1; i <= N; i++)
    {
        if (IsPerfect(i) == 1)
        {
            cout << i << " is the perfect number" << endl;

            cout << "Division of " << i << " are ";

            for (int j = 1; j <= i / 2; j++)
            {
                if (i % j == 0)
                {
                    cout << j << " ";
                }
            }
            cout << endl;
        }
    }

    return 0;
}
```

OUTPUT:

```
Enter a number: 30
6 is the perfect number
Division of 6 are 1 2 3
28 is the perfect number
Division of 28 are 1 2 4 7 14

-----
Process exited after 23.58 seconds with return value 0
Press any key to continue . . . |
```

TASK 7.3:

Functions with more than One Arguments

Write a function in C++ language that takes four integers as arguments and returns the largest of the five numbers. Your program should have the following interface.

CODE:

```
#include <iostream>
using namespace std;
int Largest(int a, int b, int c, int d)
{
    int max = a;

    if (b > max)
        max = b;
    if (c > max)
        max = c;
    if (d > max)
        max = d;

    return max;
}

int main()
{
    int a, b, c, d;

    cout << "Enter four integers" << endl;
    cin >> a;
    cin >> b;
    cin >> c;
    cin >> d;

    int result = Largest(a, b, c, d);

    cout << "The largest number is " << result;

    return 0;
}
```

OUTPUT:

```
Enter four integers
25
32
57
41
The largest number is 57
-----
Process exited after 10.8 seconds with return value 0
Press any key to continue . . . |
```

EXERCISE 7.1:

Correct the error and write down the output of the following code.

CODE:

```
#include <iostream>
using namespace std;
void generateFibonacci(int n)
{
    int first = 0, second = 1, next;

    cout << "Fibonacci Series up to " << n << " terms: ";
    for (int i = 1; i <= n; ++i)
    {
        if (i == 1)
        {
            cout << first;
        }
        else if (i == 2)
        {
            cout << ", " << second;
        }

        else
        {
            next = first + second;
            cout << ", " << next;
            first = second;
            second = next;
        }
        cout << endl;
    }
}

int main()
{
    int terms;
    cout << "Enter the number of terms for the Fibonacci series: ";
    cin >> terms;
    generateFibonacci(terms);
return 0;
}
```

OUTPUT:

```
Enter the number of terms for the Fibonacci series: 5
Fibonacci Series up to 5 terms: 0, 1, 1, 2, 3
```

```
-----
Process exited after 3.878 seconds with return value 0
Press any key to continue . . . |
```

EXERCISE 7.2:

Correct the error and write down the output of the following code.

CODE:

```
#include <iostream>
using namespace std;
bool isPrime(int num)
{
    if (num <= 1)
    {
        return false;
    }
    for (int i = 2; i * i <= num; ++i)
    {
        if (num % i == 0)
        {
            return false;
        }
    }
    return true;
}
void findAndPrintPrimes(int start, int end)

{
    cout << "Prime numbers between " << start << " and " << end << ":" ;
for (int i = start; i <= end; ++i)
{
    if (isPrime(i))
    {
        cout << i << " ";
    }
}
    cout << endl;
}
int main()
{
    int start, end;
    cout << "Enter the starting and ending numbers for the range: ";
    cin >> start >> end;
    if (start <= 1 || end <= 1 || start >= end)
    {
        cout << "Invalid range. Please enter valid starting and ending numbers." :
```

```
        |     return 1;
    }
    findAndPrintPrimes(start, end);
return 0;
}
```

OUTPUT:

```
Enter the starting and ending numbers for the range: 10 30
Prime numbers between 10 and 30: 11 13 17 19 23 29
```

```
-----
Process exited after 3.799 seconds with return value 0
Press any key to continue . . . |
```

EXERCISE 7.3:

Correct the error and write down the output of the following code.

CODE:

```
#include <iostream>
using namespace std;
unsigned long long calculateFactorial(int num)
{
    unsigned long long factorial = 1;

    for (int i = 2; i <= num; ++i)
    {
        factorial *= i;
    }
    return factorial;
}

int main()
{
    int n;
    cout << "Enter a positive integer to calculate its factorial: ";
    cin >> n;
if (n < 0 )
{
    cout << "Factorial is not defined for negative numbers." << endl;
    return 1;
}
    unsigned long long result = calculateFactorial(n);
    cout << "Factorial of " << n << " is: " << result << endl; return 0;
}
```

OUTPUT:

```
Enter a positive integer to calculate its factorial: 5
Factorial of 5 is: 120
```

```
Process exited after 2.993 seconds with return value 0
Press any key to continue . . . |
```