Assignment - 24 Job Ready Bootcamp in C++, DSA and IOT MySirG

Functions in C++

1. Define a function to check whether a given number is a Prime number or not.

Program -

```
#include <iostream>
using namespace std;

int checkPrime(int);

int main()
{
    int n;
    cout<<"Enter a number: ";
    cin>>n;

    if(checkPrime(n))
        cout<<n<<" is a Prime number";
    else
        cout<<n<<" is not a Prime number";
    return 0;
}

int checkPrime(int num)
{
    int i;
    for(i = 2; i <= num; i++)
    {
        if(num % i == 0)
            break;
    }
    if(i == num)
        return 1;
    else
        return 0;
}
```

Output -

Enter a number: 13
13 is a Prime number

---

2. Define a function to find the highest value digit in a given number.

Program -

```
#include <iostream>
using namespace std;

int highestDigit(int);
```

```
int main()
{
    int n;
    cout<<"Enter a number: ";
    cin>>n;
    cout<<"Highest value digit in "<<n<<" is "<<highestDigit(n);
    return 0;
}

int highestDigit(int num)
{
    int max = -1;
    while(num)
    {
        if(max < num % 10)
            max = num % 10;

        num = num / 10;
    }
    return max;
}
```

Output -
Enter a number: 798
Highest value digit in 798 is 9

---

3. Define a function to calculate x raised to the power y.

Program -

```
#include <iostream>
using namespace std;

int calcPow(int,int);

int main()
{
    int x, y;
    cout<<"Enter the value of x and y: ";
    cin>>x>>y;

    cout<<x<<" raised to the power "<<y<<" is "<<calcPow(x,y);
    return 0;
}

int calcPow(int a, int b)
{
    int p = 1, i = 1;

    while(i <= b)
    {
```

```
        p *= a;
        i++;
    }
    return p;
}
```

Output -
Enter the value of x and y: 5 3
5 raised to the power 3 is 125

---

4. Define a function to print Pascal Triangle up to N lines.

Program -

```
#include <iostream>
using namespace std;

void printPascal(int);
int combi(int,int);
int fact(int);

int main()
{
    int lines;
    cout<<"Enter the number of lines: ";
    cin>>lines;
    printPascal(lines);
    return 0;
}

void printPascal(int n)
{
    int i, j, k, r;

    for(i = 1; i <= n; i++)
    {
        k = 1;
        r = 0;
        for(j = 1; j <= (2*n-1); j++)
        {
            if(j >= (n+1-i) && j <= (n-1+i) && k)
            {
                cout<<combi(i-1, r);
                r++;
                k = 0;
            }
            else
            {
                cout<<" ";
                k = 1;
            }
```

```
        }
        cout<<endl;
    }
}

int combi(int n, int r)
{
    return fact(n)/(fact(r)*fact(n-r));
}

int fact(int num)
{
    int f = 1;

    while(num)
    {
        f *= num;
        num--;
    }
    return f;
}
```

Output -

Enter the number of lines: 5

```
     1
    1 1
   1 2 1
  1 3 3 1
 1 4 6 4 1
```

---

5. Define a function to check whether a given number is a term in a Fibonacci series or not.

Program -

```
#include <iostream>
using namespace std;

int checkNum(int,int);

int main()
{
    int n, terms;

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

    cout<<"Enter the number of terms for fibonacci series: ";
    cin>>terms;
```

```
    if(checkNum(n, terms))
        cout<<endl<<n<<" is present in the fibonacci series";
    else
        cout<<endl<<n<<" is not present in the fibonacci series";
    return 0;
}

int checkNum(int num, int t)
{
    int flag = 0, a = -1, b = 1, c, i;

    for(i = 1; i <= t ; i++)
    {
        c = a + b;

        if(num == c)
            flag = 1;

        cout<<c<<" ";
        a = b;
        b = c;
    }
    if(flag)
        return 1;
    else
        return 0;
}
```

Output -
Enter a number: 21
Enter the number of terms for fibonacci series: 10
0 1 1 2 3 5 8 13 21 34
21 is present in the fibonacci series

---

6. Define a function to swap data of two int variables using call by reference

Program -

```
#include<iostream>
using namespace std;

void swap(int &,int &);

int main()
{
    int a, b;

    cout<<"Enter the value of a and b: ";
    cin>>a>>b;

    cout<<"Before swapping -"<<endl;
```

```
    cout<<"a = "<<a<<", "<<"b = "<<b<<endl;

    swap(a,b);

    cout<<"After swapping -"<<endl;
    cout<<"a = "<<a<<", "<<"b = "<<b;
    return 0;
}

void swap(int &x, int &y)
{
    x = x + y;
    y = x - y;
    x = x - y;
}
```

Output -
Enter the value of a and b: 12 5
Before swapping -
a = 12, b = 5
After swapping -
a = 5, b = 12

---

7. Write a function using the default argument that is able to add 2 or 3 numbers.

Program -
```
#include<iostream>
using namespace std;

int sum(int=0, int=0, int=0);
int main()
{
    cout<<"Sum is "<<sum()<<endl;
    cout<<"Sum is "<<sum(1)<<endl;
    cout<<"Sum is "<<sum(1,2)<<endl;
    cout<<"Sum is "<<sum(1,2,3)<<endl;
    return 0;
}

int sum(int a, int b, int c)
{
    return a+b+c;
}
```

Output -
Sum is 0
Sum is 1
Sum is 3
Sum is 6

---

8. Define overloaded functions to calculate area of circle, area of rectangle and area of Triangle.

Program -

```
#include <iostream>
using namespace std;

float area(int);
float area(float,float);
float area(int,int);

int main()
{
    int r, l, b;
    float base, height;

    cout<<"Enter the radius of circle: ";
    cin>>r;
    cout<<"Area of circle: "<<area(r)<<endl;

    cout<<"Enter length and breadth of rectangle: ";
    cin>>l>>b;
    cout<<"Area of rectangle: "<<area(l, b)<<endl;

    cout<<"Enter base and height of triangle: ";
    cin>>base>>height;
    cout<<"Area of triangle: "<<area(base, height);

    return 0;
}

float area(int radius)
{
    return 3.14*radius*radius;
}

float area(int length, int breadth)
{
    return length*breadth;
}

float area(float base, float height)
{
    return (base*height)/2;
}
```

Output -

Enter the radius of circle: 6
Area of circle: 113.04
Enter length and breadth of rectangle: 8 9
Area of rectangle: 72

Enter base and height of triangle: 45 7
Area of triangle: 157.5

---

9. Write functions using function overloading to find a maximum of two numbers and both the numbers can be integer or real.

Program -

```
#include <iostream>
using namespace std;

int Max(int, int);
float Max(float, float);

int main()
{
    int a, b;
    float i, j;

    cout<<"Enter two numbers: ";
    cin>>a>>b;
    cout<<"Maximum among "<<a<<" and "<<b<<" is "<<Max(a,b)<<endl;

    cout<<"Enter two numbers: ";
    cin>>i>>j;
    cout<<"Maximum among "<<i<<" and "<<j<<" is "<<Max(i,j);
    return 0;
}

int Max(int x, int y)
{
    int res = x > y ? x : y;
    return res;
}

float Max(float p, float q)
{
    float res = p > q ? p : q;
    return res;
}
```

Output -

Enter two numbers: 4 -3
Maximum among 4 and -3 is 4
Enter two numbers: 9.4 8.9
Maximum among 9.4 and 8.9 is 9.4

---

10. Write functions using function overloading to add two numbers having different data types.

Program -

```
#include <iostream>
using namespace std;

int add(int,int);
float add(float,float);
float add(int,float);

int main()
{
    cout<<"Sum is: "<<add(3, 9)<<endl;
    cout<<"Sum is: "<<add(5.6f, 4.3f)<<endl;
    cout<<"Sum is: "<<add(45, 8.2f)<<endl;
    return 0;
}

int add(int x, int y)
{
    return x+y;
}

float add(float x, float y)
{
    return x+y;
}

float add(int x, float y)
{
    return x+y;
}
```

Output -

Sum is: 12
Sum is: 9.9
Sum is: 53.2