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

**Functions in C++**

**Abhhishek kumar**

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

#include<iostream>

using namespace std;

int prime(int n);

int main()

{

    int n;

    cout << "Enter the numbers:";

    cin >> n;

    cout << prime(n) << endl;

}

int prime(int n)

{

    int i,temp=0;

    for(i=1; i<=n; i++)

    {

        if(n%i==0)

        {

           temp++;

        }

    }

    if(temp==2)

    {

        cout << "Prime numbers:";

    }

    else

    {

        cout << "Not Prime numbers:";

    }

    return n;

}

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

#include<iostream>

using namespace std;

int High(int);

int main()

{

    int num;

    cout << "Enter the numbers:";

    cin >> num;

    cout << "Highest value of digit: " << High(num);

}

int High(int n)

{

    int r,max=0;

    while(n!=0)

    {

        r = n%10;

        if(max<r)

           max=r;

        n = n/10;

    }

    return max;

}

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

#include<iostream>

using namespace std;

int spow(int b,int p);

int main()

{

    int b,p,r;

    cout << "Enter the base:";

    cin >> b;

    cout << "Enter the power:";

    cin >> p;

    r = spow(b,p);

    cout << b << "^" << p << " = "  << r;

}

int spow(int b,int p)

{

    if(p==0)

       return 1;

    else

       return b\*spow(b,p-1);

}

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

#include<iostream>

using namespace std;

int fact(int n)

{

    int i,fact=1;

    for(i=1; i<=n; i++)

    {

        fact = fact\*i;

    }

    return fact;

}

int comb(int n,int r)

{

    return fact(n)/fact(r)\*fact(n-r);

}

int par(int n,int r)

{

    return fact(n)/(fact(n-r));

}

int pascal(int n)

{

    int i,j;

    for(i=0; i<=n; i++)

    {

        for(j=1; j<=i; j++)

        {

            cout << comb(i,j);

        }

        cout << endl;

    }

}

int main()

{

    cout << "Combination is " << pascal(3);

}

/\*5. Define a function to check whether a given number is a term in a Fibonacci series or

not.\*/

#include<iostream>

using namespace std;

int fib(int);

int main()

{

    int n;

    cout << "Enter the numbers:";

    cin >> n;

    fib(n);

}

int fib(int n)

{

    int a=0,b=1,c;

    c = a+b;

    while(c<n)

    {

        a = b;

        b = c;

        c = a+b;

    }

    if(c==n)

    {

        cout << "Fibonacci:";

    }

    else

    {

        cout << "Not Fibonacci:";

    }

}

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

#include<iostream>

using namespace std;

int swap(int \*,int \*);

int main()

{

    int a,b;

    cout << "Enter the two numbers:";

    cin >> a >> b;

    cout << "Before swap" << " " << "a=" << a << " " << "b=" << b << endl;

    swap(&a,&b);

    cout << "After swap" << "  " << "a=" << a << " " << "b=" << b;

}

int swap(int \*p,int \*q)

{

    int temp;

    temp = \*p;

    \*p = \*q;

    \*q = temp;

}

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

#include<iostream>

using namespace std;

int add(int,int);

int add(int,int,int);

int main()

{

    int a,b,c;

    cout << "Enter the two numbers:";

    cin >> a >> b;

    cout << "Add two numbers: "<< add(a,b) << endl;

    cout << "Enter the third numbers:";

    cin >> c;

    cout << "Add three numbers: "<< add(a,b,c) << endl;

}

int add(int x, int y)

{

    return x+y;

}

int add(int x, int y, int z)

{

    return x+y+z;

}

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

triangle

#include<iostream>

using namespace std;

float Area(float);

int area(int , int);

int ar\_ea(int,int);

int main()

{

    float r;

    int b,h,bs,he;

    cout << "Enter the radius:";

    cin >> r;

    cout << "Enter the base and height of rectangle:";

    cin >> b >> h;

    cout << "Enter the base and height of triangle:";

    cin >> bs >> he;

    cout << endl;

    cout << "Area of Circle: " << Area(r) << endl;

    cout << "Area of Rectangle: " << area(b,h) << endl;

    cout << "Area of Triangle: " << ar\_ea(bs,he);

}

float Area(float r)

{

    return 3.14\*r\*r;

}

int area(int b,int h)

{

    return b\*h;

}

int ar\_ea(int bs, int he)

{

    return (bs\*he)/2;

}

/\*9. Write functions using function overloading to find a maximum of two numbers and

both the numbers can be integer or real.\*/

#include<iostream>

using namespace std;

int maximum(int,int);

int main()

{

    int a,b;

    cout << "Enter the two numbers:";

    cin >> a >> b;

    cout << "Maximum is :"<< maximum(a,b) << endl;

}

int maximum(int x,int y)

{

    if(x>y)

       return x;

    else

       return y;

}

/\*10. Write functions using function overloading to add two numbers having different data

types\*/

#include<iostream>

using namespace std;

int add(int,int);

int main()

{

    int a,b;

    cout << "Enter the two numbers:";

    cin >> a >> b;

    cout << "Add two numbers: "<< add(a,b) << endl;

}

int add(int x, int y)

{

    return x+y;

}