Q1. Define a function to check whether a given number is a Prime number or not.

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

int prime(int num);

int main() {

	int num;

	cout << "Enter number = ";
	cin >> num;

	if (prime(num))
		cout << num << " is a prime number";
	else
		cout << num << " is not a prime number";

	return 0;

}
int prime(int num) {
	for (int i = 2; i < num; i++) {

		if (num % i == 0)
			return 0;
```

```
        }
        return 1;
}
```

Q2. Define a function to find the highest value digit in a given number.

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

int highest(int num);

int main() {
	int num;

	cout << "Enter number = ";
	cin >> num;

	cout << "Heighest number = " << highest(num);
}
int highest(int num) {

	int digit, h = 0;

	while (num != 0) {

		digit = num % 10;
		num = num / 10;

		if (h < digit)
			h = digit;
	}
```

```
        return h;
}
```

Q3. Define a function to calculate x raised to the power y.

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

void calculate(void);

int main() {

	calculate();
}
void calculate(void) {
	int x, y, ans = 1;

	cout << "Enter number = ";
	cin >> x;

	cout << "Enter power = ";
	cin >> y;

	while (y) {
		ans = ans * x;
		y--;
	}
	cout << "\n Answer = " << ans;
}
```

Q4. Define a function to print Pascal Triangle up to N lines.

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

int fact(int num);
int nCr(int n, int r);
void PascalTriangle(int row);

int main() {

	int row;

	cout << "Enter row : ";
	cin >> row;

	PascalTriangle(row);
}

int fact(int num) {

	int ans = 1;

	for (int i = 2; i <= num; i++)
		ans = ans * i;

	return ans;
```

```
}

int nCr(int n, int r) {

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

void PascalTriangle(int row) {
	int space = row;

	for (int i = 0; i < row; i++) {
		for (int s = 1; s < space; s++) {
			cout << " ";
		}
		for (int j = 0; j <= i; j++) {
			int ncr = nCr(i, j);

			if (ncr <= 9)
				printf("%2d ", ncr);
			else
				printf("%d ", ncr);
		}
		printf("\n");
		space = space - 1;
	}
}
```

Q5. Define a function to check whether a given number is a term in a Fibonacci series or not.

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

void chkFibonacci(void);

int main() {

	chkFibonacci();

}

void chkFibonacci(void) {

	int prv = 0, nxt = 1, num;

	cout << "Enter number : ";
	cin >> num;

	while (num > nxt) {

		cout << prv + nxt << " ";

		nxt = prv + nxt;
		prv = nxt - prv;
```

```
    }
    cout << endl << endl;
    if (num == nxt)
        cout << "Number is in Fibonacci";
    else
        cout << "Number is not in Fibonacci";
}
```

Q6. Define a function to swap data of two int variables using call by reference.

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

void swap(int *a, int *b);

int main() {

	int a = 2, b = 3;

	cout << "Before Swapping : a = " << a << "   b = " << b;
	swap(&a, &b);
	cout << "\nAfter Swapping : a = " << a << "   b = " << b;

}

void swap(int *a, int *b) {

	*a = *a + *b;
	*b = *a - *b;
	*a = *a - *b;
}
```

Q7. 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 = 0);

int main() {

	cout << add(5, 5) << endl;
	cout << add(5, 5, 5);

}

int add(int a, int b, int c) {

	return a + b + c;
}
```

Q8. Define overloaded functions to calculate area of circle, area of rectangle and area of triangle.

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

float area(float r);
int area(int l, int b, int h);
double area(double b, double h);

int main() {

    float r;
    int l, w;
    double base, height;

    cout << "Enter radius = ";
    cin >> r;

    cout << "Enter Length,Width = ";
    cin >> l >> w;

    cout << "Enter base, height = ";
    cin >> base >> height;

    cout << endl << "Area of circle = " << area(r);
    cout << endl << "Area of rectangle = " << area(l, w);
```

```
        cout << endl << "Area of triangle = " << area(base, height);

}

float area(float r) {
        return 3.14 * ( r * r);
}

int area(int l, int w) {
        return l * w;
}

double area(double b, double h) {
        return (h * b) / 2;
}
```

Q9. 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 max(int a, int b);
float max(float c, float d);

int main() {

	int a, b;
	float c, d;

	cout << "Enter number = ";
	cin >> a >> b;
	cout << "Maximum number = " << max(a, b) << endl << endl;

	cout << "Enter number = ";
	cin >> c >> d;
	cout << "Maximum number = " << max(c, d) << endl << endl;
}

int max(int a, int b) {
	if (a > b)
		return a;
	else
```

```
        return b;
}

float max(float c, float d) {
    if (c > d)
        return c;
    else
        return d;
}
```

Q10. Write functions using function overloading to add two numbers having different data types.

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

int add(int a, int b);
float add(float c, float d);

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

	cout << "Enter number = ";
	cin >> a >> b;
	cout << "Sum is = " << add(a, b);

	cout << endl << endl << "Enter number = ";
	cin >> c >> d;
	cout << "Sum is = " << add(c, d);
}
int add(int a, int b) {
	return a + b;
}
float add(float c, float d) {
	return c + d;
}
```