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.

int primeNum(int n)

{

bool flag = false;

for (int i = 2; i <= n / 2; i++)

{

if (n % i == 0)

{

flag = true;

break;

}

}

if(flag)

return 1; // primeNo.

else

return 0; // noPrimeNo.

// if(flag)

// cout<<"Prime Number!";

// else

// cout<<"Not Prime Number!";

}

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

#include <iostream>

using namespace std;

int heighestValueDigit(int digit)

{

int maxDigit = -1;

while (digit != 0)

{

if (maxDigit < digit % 10)

maxDigit = digit % 10;

digit /= 10;

}

return maxDigit;

}

int main()

{

cout << heighestValueDigit(572);

return 0;

}

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

#include <iostream>

using namespace std;

int power(int x, int y)

{

int ans = 1;

for (int i = 1; i <= y; i++)

ans \*= x;

return ans;

}

int main()

{

cout << power(5, 5);

return 0;

}

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

#include <iostream>

using namespace std;

int fact(int n)

{

int ans = 1;

for (int i = 2; i <= n; i++)

ans = ans \* i;

return ans;

}

int nCr(int n, int r)

{

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

}

void pascalTriangle(int lines)

{

for (int i = 1; i <= lines; i++)

{

int k = 1, col = 0;

for (int j = 1; j < lines \* 2; j++)

{

if (j >= (lines + 1) - i && j <= (lines - 1) + i && k)

{

cout << nCr(i - 1, col) << " ";

col++;

k = 0;

}

else

{

cout << " ";

k = 1;

}

}

cout << endl;

}

}

int main()

{

pascalTriangle(5);

return 0;

}

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

not.

#include <iostream>

using namespace std;

bool checkFibonacciSeriesNum(int num)

{

if (num == 0 || num == 1)

return true;

else

{

int x = 0, y = 1, temp;

while (x + y <= num)

{

if (x + y == num)

return true;

temp = x + y;

x = y;

y = temp;

}

}

return false;

}

int main()

{

int num;

cout << "Enter a number: ";

cin >> num;

if (checkFibonacciSeriesNum(num))

cout << num << " is present in fibonacci series!";

else

cout << num << " is not present in fibonacci series!";

return 0;

}

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

#include <iostream>

using namespace std;

void swap(int &, int &);

int main()

{

int num1, num2;

cout << "Enter two number: ";

cin >> num1 >> num2;

cout << "\nBefore swaping: ";

cout << "Num1 = " << num1 << " and Num2 = " << num2 << endl;

swap(num1, num2);

cout << "\nAfter swaping: ";

cout << "Num1 = " << num1 << " and Num2 = " << num2 << endl;

return 0;

}

void swap(int &x, int &y)

{

int temp = x;

x = y;

y = 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 num1, int num2, int num3 = 0); // this function able to add too argument or three argument number

int main()

{

int num1, num2, num3;

cout << "Enter two number: ";

cin >> num1 >> num2;

cout << "Sum of " << num1 << " and " << num2 << " is: " << add(num1, num2) << endl;

cout << "\nNow enter three number: ";

cin >> num1 >> num2 >> num3;

cout << "Sum of " << num1 << ", " << num2 << " and " << num3 << " is: " << add(num1, num2, num3) << endl;

return 0;

}

int add(int num1, int num2, int num3)

{

return num1 + num2 + num3;

}

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

Triangle

#include <iostream>

using namespace std;

float area(int r)

{

return 3.14 \* r \* r;

}

float area(int w, int h)

{

return w \* h;

}

float area(double b, double h)

{

return ((b \* h) / 2.0);

}

int main()

{

cout << "Area of Circle: " << area(5) << ", having " << 5 << " radius" << endl;

cout << "Area of Rectangle " << area(3, 4) << ", having " << 3 << " width and " << 4 << " height" << endl;

cout << "Area of Trinalge " << area(3.5, 4.3)<< ", having " << 3 << " breadth and " << 4 << " height" << endl;;

return 0;

}

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 heighestValueDigit(int digit)

{

int maxDigit = -1;

while (digit != 0)

{

if (maxDigit < digit % 10)

maxDigit = digit % 10;

digit /= 10;

}

return maxDigit;

}

int main()

{

cout << heighestValueDigit(572);

return 0;

}

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

types.

#include <iostream>

using namespace std;

// we can make different type of function overloading for diff data type here we make only three function for int, float and double, you can also make for sort, long, long long etc.

int add(int num1, int num2)

{

return num1 + num2;

}

float add(float num1, float num2)

{

return num1 + num2;

}

double add(double num1, double num2)

{

return num1 + num2;

}

using namespace std;

int main()

{

double num1, num2;

cout << "Enter two int type number: ";

cin >> num1 >> num2;

cout << " Sum is: " << add((int)num1, (int)num2) << endl;

cout << "Enter two float type number: ";

cin >> num1 >> num2;

cout << "Sum is: " << add((float)num1, (float)num2) << endl;

cout << "Enter two double type number: ";

cin >> num1 >> num2;

cout << "Sum is: " << add(num1, num2) << endl;

return 0;

}