

Computer Programming (CS1123) – (Section 2)

<pre>void fun1(); void fun2(); void main(){ cout<<"A"<<endl; fun1(); cout<<"B"<<endl; fun2(); cout<<"C"<<endl; } void fun1() { cout<<"fun1"<<endl; } void fun2() { cout<<"fun2"<<endl; }</pre>	<pre>void fun1(); void fun2(); void main(){ cout<<"A"<<endl; fun1(); cout<<"B"<<endl; cout<<"C"<<endl; } void fun1(){ cout<<"fun1"<<endl; fun2(); } void fun2(){ cout<<"fun2"<<endl; }</pre>
<pre>void fun1(); void fun2(); void main(){ cout<<"A"<<endl; fun1(); cout<<"B"<<endl; fun2(); cout<<"C"<<endl; } void fun1(){ cout<<"fun1"<<endl; fun1(); } void fun2(){ cout<<"fun2"<<endl; fun1(); }</pre>	<pre>void fun1(); void fun2(); void main(){ cout<<"A"<<endl; fun1(); cout<<"B"<<endl; fun2(); cout<<"C"<<endl; } void fun1(){ cout<<"fun1"<<endl; fun2(); } void fun2(){ cout<<"fun2"<<endl; fun1(); }</pre>
<pre>int square(int &sq){ sq = sq * sq; return sq; } void main(){ int a = 5; cout<<a<<"\ Square ="<<square(a); }</pre>	<pre>int square(int sq){ sq = sq * sq; return sq; } void main(){ int a = 5; cout<<a<<"\ Square ="<<square(a); }</pre>
<pre>int MFunc(int m){ m = m * m; return m; } void main(){ int temp, var1 = 5, var2 = 5; temp = MFunc(var1+var2); }</pre>	<pre>int MFunc(int m){ m = m * m; return m; } void main(){ int temp, var1 = 5, var2 = 5; temp = MFunc(var1+var2); cout<<temp; }</pre>

<pre>int Function(int m1, int m2){ cout<<m1<<endl<<m2<<endl; return m1 * m2; } void main(){ int n1 = 2, n2 = 5, n3 = 7, n; n = Function(Function(n1,n3),n2); cout<<n<<endl; }</pre>	<pre>int Function(int m1, int m2){ return m1 * m2; } void main(){ int n; n = Function(Function(5,2),5); cout<<n<<endl; }</pre>
<pre>int computation(int m1, int m2) { int sum = 0; for(int i=m1; i<m2; i++) sum += m1 + m2; return sum; } void main(){ int n = computation(5,7); cout<<n<<endl; }</pre>	<pre>int a = 10; int b = 70; void Test(){ a = a + b; int b = a++; int c = b; } void main(){ int a = 2; int b = 4; Test(); cout<<a<<" "<<b<<" "<<endl; }</pre>
<pre>int a = 10; int b = 70; void Test(){ a = a + b; int b = a++; int c = b; } void main(){ int a = 2; int b = 4; Test(); cout<<::a<<" "<<::b<<" "<<endl; }</pre>	<pre>int a = 10; int b = 70; void Test(){ a = a + b; int b = a++; int c = b; } void main(){ int a = 2; int b = 4; cout<<:::a<<" "<<:::b<<" "<<endl; }</pre>
<pre>int factorial(int); void main(void) { int number = 3; cout<<factorial(number)<< endl; } int factorial(int n) { int temp; if(n <= 1) return 1; else temp = n * factorial(n - 1); return temp; }</pre>	<pre>int factorial(int); void main(void) { int number = 5; cout<<factorial(number)<< endl; } int factorial(int n) { int temp; temp = n * factorial(n - 1); return temp; }</pre>

<pre> Int ArrSum(int, int, int[], int[]); void main(){ int arr1[3] = {10,20,30}; int arr2[3] = {5,10,15}; cout<<ArrSum(0, 3, arr1, arr2); } int ArrSum(int s, int f, int a1[], int a2[]) { int sum = 0; for(int i=s; i<f; i++) sum += a1[i] + a2[i]; return sum; } </pre>	<pre> int Test(int a, int &b) { while(a < b){ a++; b = b - 1; } if(a >= 5) return a; else return b; } void main(){ int x = 5, y = 10; int z = Test(x,y); cout<<x<<" "<<y<<" "<<z; } </pre>
--	--

Q2: Write a function to display a 2-D array of size 50 x 50 in matrix form. Declare and Initialize that array within the function.

Q3: Write a function to display a 2-D array of size 3 x 3 in matrix form. Pass 2-D array from main() function.

```

void Matrix(int arr[][3],const int size){
    for(int i=0; i<size; i++){
        for(int j=0; j<size; j++)
            cout<<arr[i][j]<<" ";
        cout<<endl;
    }
}
void main(){
    int mArr[][3]={{1,2,3},{1,2,3},{1,2,3}};
    Matrix(mArr,3);
}

```

Q4: Write a function which confirms that the passed number is prime number or not.

Q5: Write a function named Sort(int arr[]) to sort elements of array in ascending order. Pass 1-D array from main() function.