Programming Fundamentals

Assignment No 6

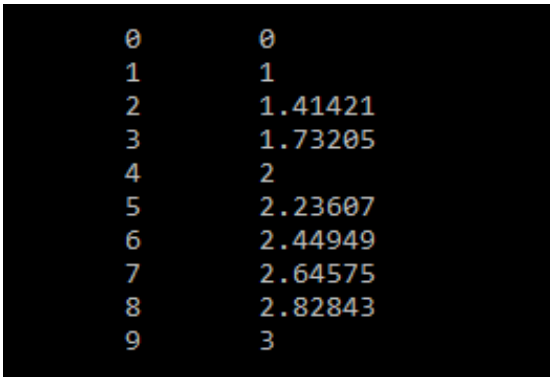
Student Name : MOMIN HAYAT KHAN
Roll No: S20-0273
Department : BS(Artificial Intelligence)
Batch / Year : SPRING 2020
Lecturer: Mam Misbah

Assignment No 6

Program No 1:

```
#include <cmath> // defines the sqrt() function
#include <iostream>
using namespace std;
int main()
{ // tests the sqrt() function:
for (int x=0; x <=9; x++)
cout << "\t" << x << "\t" <<sqrt(x)<<endl;}
```

Output:



Types of Function:

There are two types of functions.

- User Defined function.
- Pre Defined function.

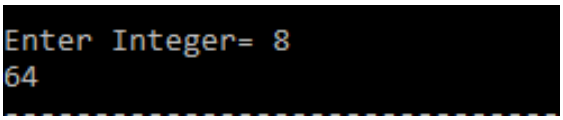
Syntax Of function (Prototype, call & body of function):

```
// Create a function
void myFunction() {
  cout << "I just got executed!";
}

int main() {
  myFunction(); // call the function
  return 0;}
```

Program No 2:

```
#include <iostream>
using namespace std;
void sqr(int a){
    cout<<a*a;
}
int main()
{   int a;
    cout<<"Enter Integer= ";cin>>a;
    sqr(a);}
```

Output:



Program No 3:

```
#include <iostream>
using namespace std;
void average1(float a,float b, float c){
    float sum=a+b+c;
    float ave=(sum*100)/300;
    cout<<ave<<"%";}
int main()
{       float a,b,c;
    cout<<"Enter three numbers for Average= ";cin>>a>>b>>c;
    average1(a,b,c);}
```

Output:

```
Enter three numbers for Average= 98
77
88
87.6667%
--------------------------------
```

Program No 4:

```
#include <iostream>
using namespace std;
int cube(int n)
{return n*n*n;}
int main()
{   int r;
    do{int n;
    cout<<"Input a number= ";cin>>n;
    cube(n);
    cout<<cube(n);
    cout<<"\nEnter 0 to terminate program= ";cin>>r;
    } while (r!=0);}
```

Output:

```
Input a number= 5
125
Enter 0 to terminate program= 1
Input a number= 7
343
Enter 0 to terminate program= 0
```

Program No 5:

```
#include <iostream>
using namespace std;
int average(int l,int w)
{return l*w;}
int main()
{   int l,w;
    cout<<"Input length= ";cin>>l;
    cout<<"Input Width= ";cin>>w;
    cout<<"average= "<<average(l,w);}
```

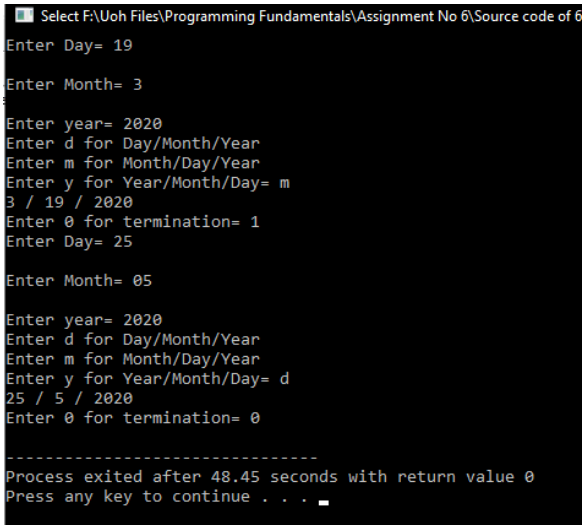
Output:

```
Input length= 25
Input Width= 6
average= 150
```

Program No 6:

```
#include <iostream>
using namespace std;
void printdate(int day=0,int month=0,int year=0)
{   int termi;
    do{
    int day,month,year;
    char select;
    cout<<"Enter Day= ";
    cin>>day;
    cout<<"\nEnter Month= ";
    cin>>month;
    cout<<"\nEnter year= ";
    cin>>year;
    if (day>=1 && day<=31 && month>0 && month<13 && year>0)
    {
                cout<<"Enter d for Day/Month/Year\nEnter m for Month/Day/Year\nEnter y for
Year/Month/Day= ";
                cin>>select;
                switch (select)
                {
                case 'd':
                    cout<<day<<" / "<<month<<" / "<<year;
                    break;
                case 'm':
                    cout<<month<<" / "<<day<<" / "<<year;
                    break;
                case 'y':
                    cout<<year<<" / "<<month<<" / "<<day;
                    break;

                default:
                    cout<<"You entered invalid character!";
                    break;
                }
    }
    else
    {
        cout<<"You Entered invalid Date";
    }

    cout<<"\nEnter 0 for termination= ";
    cin>>termi;
    }while (termi!=0);
}
int main()
{
    printdate();
}
```

Output:

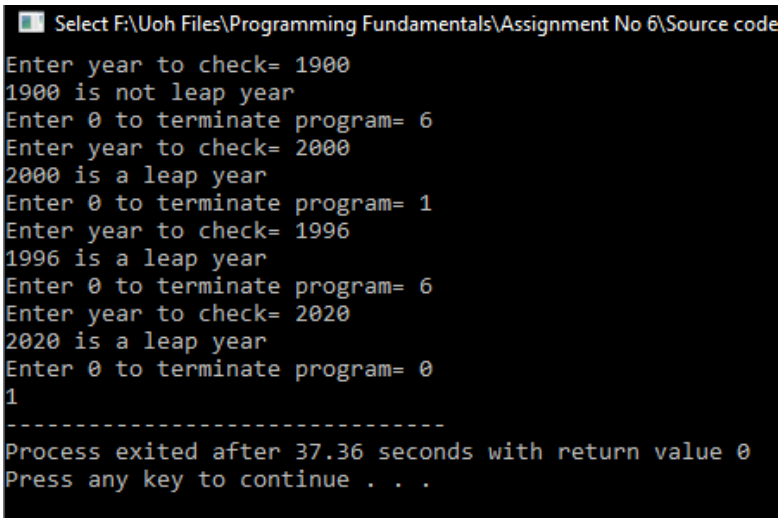


Program No 7:

```
#include <iostream>
using namespace std;
bool isleapyear(int year=0)
{   int terim;
    do
    {
    cout<<"Enter year to check= ";
    cin>>year;
    if(year%100==0)
    {
        if (year%400==0)
        {
            cout<<year<<" is a leap year";
        }
       else
       {
           cout<<year<<" is not leap year";
       }

    }
    else if (year%400)
    {
        cout<<year<<" is a leap year";
    }
    else
    {
        cout<<year<<" is not leap year";
    }

    cout<<"\nEnter 0 to terminate program= ";
    cin>>terim;
    } while (terim!=0);
    return year;

}
int main()
{
```

```
   cout<<isleapyear();

}
```

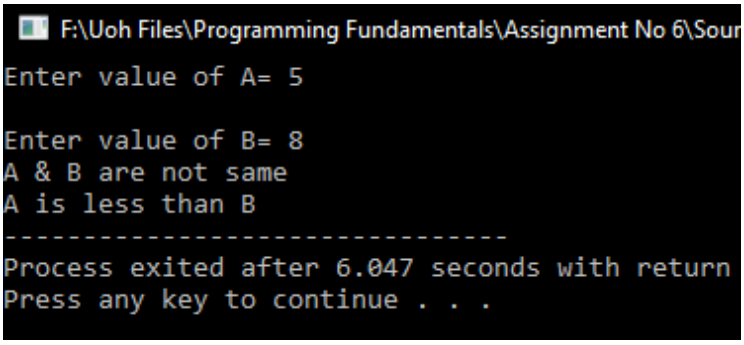
Output:



Program No 8:

```
#include <iostream>
using namespace std;
void tocheck(int a,int b)
{
    if(a==b)
    {
        cout<<"A  B are the same";
    }
    else if(a<b)
    {
        if(a!=b){
        cout<<"A & B are not same";
        cout<<"\nA is less than B";
    }}
   else
   {
      cout<<"A is Greater than B";
   }
}

int main()
{
    int a,b;
    cout<<"Enter value of A= ";
    cin>>a;
    cout<<"\nEnter value of B= ";
    cin>>b;
    tocheck(a,b);
}
```

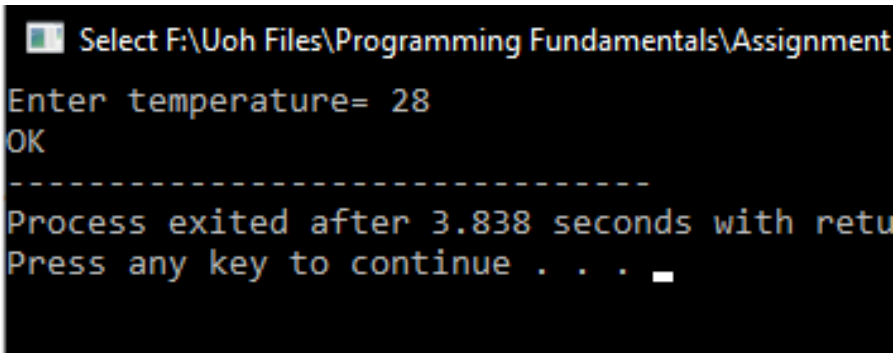
Output:



Program No 9:

```
#include <iostream>
using namespace std;
void printTempOpinion(int temp)
{
   if(temp<=10)
   {
      cout<<"COLD";
   }
   else if(temp>=20 && temp<=30)
   {
      cout<<"OK";
   }
   else if(temp>30)
   {
     cout<<"HOT";
   }

}
int main()
{
   int checktep;
   cout<<"Enter temperature= ";
   cin>>checktep;
   printTempOpinion(checktep);
}
```

Output:



Comparison of functions

Difference between Passing By Value Versus Passing By Reference

By value	By Refrence
While calling a function, we pass values of variables to it. Such functions are known as "Call By Values".	While calling a function, instead of passing the values of variables, we pass address of variables(location of variables) to the function known as "Call By References.
In this method, the value of each variable in calling function is copied into corresponding dummy variables of the called function.	In this method, the address of actual variables in the calling function are copied into the dummy variables of the called function.
With this method, the changes made to the dummy variables in the called function have no effect on the values of actual variables in the calling function.	With this method, using addresses we would have an access to the actual variables and hence we would be able to manipulate them.

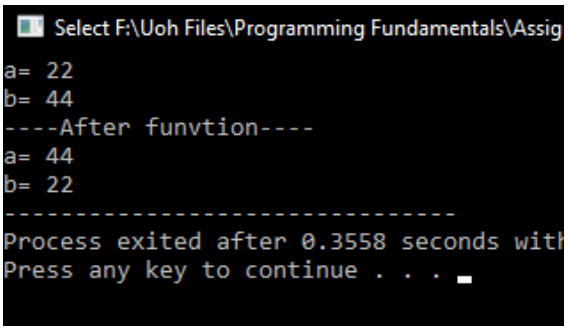
Program:
#include <iostream>
using namespace std;
void swapNum(int n)
{
n+=10;
cout<<n;
}
int main()
{
int n;
n=10;
cout<<n<<endl;
swapNum(n);
cout<<"\nAfter function= ";
cout<<n;
}
 	
Output:
10
20
After function= 10
	Program:
	#include <iostream>
	using namespace std;
	void swapNum(int &n)
	{
	n+=10;
	cout<<n;
	}
	int main()
	{
	int n;
	n=10;
	cout<<n<<endl;
	swapNum(n);
	cout<<"\nAfter function= ";
	cout<<n;
	}
	
	Output:
	10
	20
	After function= 20

Program No 10:

```
#include <iostream>
using namespace std;
void swapNum(int &a,int &b)
{
  a=44.4;
  b=22.2;
}
int main()
{
  int a=22.2;
  int b=44.4;
  cout<<"a= "<<a<<endl;
  cout<<"b= "<<b<<endl;
  swapNum(a,b);
  cout<<"----After funvtion----"<<endl;
```

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

}
```

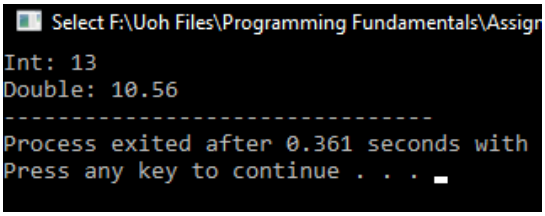
Output:



Program No 11:

```
#include <iostream>
using namespace std;
int plusFunc(int x, int y) {
  return x + y;
}

double plusFunc(double x, double y) {
  return x + y;
}

int main() {
  int myNum1 = plusFunc(8, 5);
  double myNum2 = plusFunc(4.3, 6.26);
  cout << "Int: " << myNum1 << "\n";
  cout << "Double: " << myNum2;
  return 0;
}
```

Output:



Program No 12:

```
#include <iostream>
using namespace std;
void factorial()
{
  int f;
  for (int n = 1; n < 11; n++)
  {
    int f=1;
```

```
  for(int j=1;j<=n;j++)
  {

    f=f*j;
  }
  cout<<"Factorial of "<<n<<"="<<f<<endl;
}
}
  int main()
  {
    factorial();
  }
```

Output: