Q1. Write a C++ program to demonstrate the addition of multiple types of data using generic functions or templates.

```
#include <iostream>
#include <fstream>
#include <conio.h>
using namespace std;

template <typename A, typename B, typename C>
C add(A a, B b);

template <typename T>
T ans;

int main()
{
   ans<int> = add<int, int, int>(5, 4);
   cout<<"Addition (int) = "<<ans<int><<endl;

   ans<float> = add<float, float, float>(5.5f, 5.2f);
   cout<<"Addition (float) = "<<ans<float><<endl;

   ans<double> = add<int, float, double>(7, 5.1);
   cout<<"Addition (double) = "<<ans<double><<endl;
}

template <typename A, typename B, typename C>
C add(A a, B b)
{
   C c = a + b;

   return c;
}
```

Q2. Write a C++ Program to find Largest among two numbers using function template.

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

template <typename X>
X Large(X a, X b);

int main()
{
   cout<<"float = "<<Large<float>(5, 5.6f)<<endl;
   cout<<"int = "<<Large<int>(5.5f, 6);
}

template <typename X>
X Large(X a, X b)
{
   return (a>b)? a: b;
}
```

Q3. Write a C++ program to find the largest of three elements using a template.

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

template <typename A = int, typename B = int, typename C = int, typename X = int>
X Large(A a, B b, C c)
   {
      if(a > b)
      {
         if(a > c)
         {
            return a;
         }
         else
         {
            return c;
         }
      }
      else
      {
         if(b > c)
         {
            return b;
         }
         else
         {
            return c;
         }
      }
   }

int main()
{
   cout<<"int float double = "<<Large<int, float, double, double>(5, 6.5f, 7.5)<<endl;
   cout<<"int float double = "<<Large<int, float, double, int>(5, 6.5f, 7.5)<<endl;
   cout<<"int float double = "<<Large(5, 6.5f, 7.5)<<endl;
}
```

Q4. Write a C++ Program to Swap data using function template.

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

template <typename T>
void Swap(T a, T b)
{
   cout<<"Before swap"<<endl<<"a = "<<a<<endl<<"b = "<<b<<endl;

   T c;

   c = a;
   a = b;
   b = c;

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

int main()
{
   Swap<int>(10, 20);
   Swap<float>(50.5f, 60.9f);
}
```

Q5. Write a C++ Program to Add two numbers using function template.

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

template <typename T>
T add( T a, T b);

int main()
{
   cout<<"int = "<<add<int>(5, 5)<<endl<<endl;
   cout<<"float = "<<add<float>(5.5f, 5)<<endl<<endl;
   cout<<"double = "<<add<double>(5.6, 5.6)<<endl<<endl;
}

template <typename T>
T add( T a, T b)
{
   return a + b;
}
```

Q6. Write a C++ Program to find Sum of Array using function template.

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

int main()
{
   int n = 0, sum = 0, p = 0;

   cout<<"How many elements you want to enter = ";
   cin>>n;

   vector<int> v(n);

   cout<<"Enter element in v"<<endl<<endl;

   for(int i = 0; i < n; i++)
   {
      cout<<i<<") position = ";
      cin>>p;

      v.insert(v.begin()+i, p);
   }

   for(int i = 0; i < v.size(); i++)
   {
      sum = sum + v.at(i);
   }

   cout<<"Addition = "<<sum;
}
```

Q7. Write a C++ Program of Templated class derived from Non-templated class.

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

class Parent
{
public:
   int number()
   {
      return 5;
   }

};

template <typename T>
class Child: public Parent
{
public:
   void addition(T num)
   {
      cout<<"Addition = "<<number() + num;
   }
};

int main()
{
   Child<float> c;
   c.addition(10.2);
}
```

Q8. Write a C++ Program to implement push and pop methods from stack using template.

```
#include <iostream>
#include <conio.h>
#include <vector>
using namespace std;

int main()
{
   vector<int> v = {10, 20, 30, 40, 50, 60};

   int ch = 0;

   while(ch != 3 )
   {
      cout<<"1. Push element"<<endl;
      cout<<"2. Pop element"<<endl;
      cout<<"3. Exit"<<endl<<endl;

      cout<<"Enter choice >>> ";
      cin>>ch;

      switch(ch)
      {
      case 1:
         {
            int n = 0;

            cout<<"Enter element to push = ";
            cin>>n;

            v.push_back(n);

            cout<<"Push Successfull"<<endl<<endl;

            for(int n:v)
               cout<<n<<" ";

            getch();

            break;

            cout<<endl<<endl;
         }
      case 2:
         {
            v.pop_back();
```

```
            for(int n:v)
               cout<<n<<" ";

            cout<<"Pop Successfull"<<endl<<endl;

            getch();

            break;

            cout<<endl<<endl;
         }
    }

  }
}
```

Q9. Write a C++ Program to Perform Simple Addition Function Using Templates.

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

template <typename T>
void add()
{
   T num1, num2, ans;

   cout<<"Enter two number = ";
   cin>>num1>>num2;

   ans = num1 + num2;

   cout<<"Answer  =  "<<ans;
}

int main()
{
   cout<<endl<<"Int data type"<<endl;
   add<int>();
   cout<<endl<<"Float data type"<<endl;
   add<float>();

}
```

Q10. Write a C++ program to implement Hash Table using Template Class.

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

int main()
{
   map <int, string> m;

   m.insert({3, "Ramteke"});
   m[1] = "yash";
   m.insert({2, "vrushabh"});

   cout<<"size = "<<m.size()<<endl;

   for(map<int, string>::iterator i = m.begin(); i != m.end(); i++)
   {
      cout<<i->first<<" : "<<i->second<<endl;
   }
}
```