

6. (6points) Insert C++ codes for generic implementation of the display function below. Assume all necessary header files are already included.

// Put your code here for generic implementation of the "display" function. Only one display function should be defined.

```
void main()
{
    list<int> li;
    vector<double> vs;
    li.push_back(3); li.push_back(7); li.push_back(10);
    vs.push_back(5.21); vs.push_back(2.1); vs.push_back(5.3);
    cout<< "li: "; display(li.begin(), li.end()); cout<< endl;
    cout<< "vs: "; display(vs.begin(), vs.end()); cout<< endl;
}
```

Execution Output result:

```
li: 3 7 10
vs: 5.21 2.1 5.3
```

7. (10points) Consider following C++ code and its execution output result.

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

```
class Student
{
public :
    Student(int id , char* name)
    {
        stu_id = id;
        stu_name = name;
    }
}
```

(a)

```
char* getName() { return stu_name; }
int getID() { return stu_id; }
```

```
private :
    int stu_id; // student id
    char* stu_name; // student name
};
```

```
int main()
{
```

```
    vector<Student> stu_vec;
```

```
    stu_vec.push_back(Student(4,"Nancy"));
    stu_vec.push_back(Student(1,"Tom"));
    stu_vec.push_back(Student(3,"Mike"));
    stu_vec.push_back(Student(2,"Lisa"));
```

```
    sort(stu_vec.begin() , stu_vec.end()); // sort by student id
```

(b)

```
// code for printing out each element (student id , name) of "stu_vec"
```

```
    return 0;
}
```

Execution Output result:

```
1 : Tom
2 : Lisa
3 : Mike
4 : Nancy
```

(1) (5points) What code should be inserted in (a) ?

//Put your code here.

(2) (5points) What code should be inserted in (b) ?

//Put your code here for printing out each element [student id : name] of "stu_vec".