

❖ Tìm hiểu và cài đặt các lớp *<template>* có dạng cho sẵn bên dưới sao cho các chức năng hoạt động phù hợp:

```
#include <iostream>
#include <fstream>
#include <cstring>
#include <algorithm>
using namespace std;
template <typename T>
class Mang
{
    public:
        Mang();
        Mang(T * data, int n);
        ~Mang();
        void sort();
        int timkiem(const T & key) const;
        void ghifile(char * tenfile);
        void docfile(char * tenfile);
        int timmax()const;
        int timmin()const;
        void xuat()const;
        void nhap();
        bool ktRong()const;
    private:
        T * data;
        int n;
};
```

Ứng dụng lớp *Mang<template>* phía trên để xây dựng một chương trình quản lý sinh viên, dùng *Mang<template>* để lưu trữ. Biết *class SinhVien* và hàm *Main* được mô tả như sau:

```
class Hocsinh
{
    public:
        Hocsinh(char hoten1[], int tuoi, float dtb)
        {
            ...
        }
        friend ostream & operator << (ostream & os, Hocsinh & r);
        friend istream & operator >> (istream & is, Hocsinh & r);
        bool operator > (Hocsinh & r)
        {
            ...
        }
        bool operator < (const Hocsinh & r);
        friend void sapxep(Hocsinh a[], int n);
    private:
        char hoten[55];
        int tuoi;
        float dtb;
};
```

```

int main()
{
    int  a1[] = {10,  2,  13,  24};
    string  s1[] = {"Kim  Tan",  "Bi  Rain",  "Le  Roi",  "Tran  Trang"};
    int  n1 = 4;

    Mang<int>  mi;
    Mang<string>  ms;
    Mang<Hocsinh>  mhs;
    mhs.nhap();
    mi.nhap();
    ms.nhap();
    mhs.ghifile("hocsinh.txt");
    mhs.sort();
    ms.sort();
    mi.sort();
    ms.xuat();
    mi.xuat();
    mhs.xuat();
    int  kq = ms.timkiem(string("Teo"));
    if (kq >= 0)
        cout <<  "Tim  thay  Teo  o  vi  tri  " <<  kq <<  endl;
    else
        cout <<  "Khong  tim  thay  Teo" <<  endl;
    return 0;
}

```