```
June Exam 2018

//Question 4
//4.1

#ifendef WEATHER_H INCLUDED
#define WEATHER_H INCLUDED
#include <iostream>
#include <istream>

using namespace std;

class Weather
{
      public:
        Weather();
        Weather(string dat, int ht, int lt, int rf);
        ~Weather();
        int getRainFall() const;
        friend bool operator >(const Weather& weather1, const Weather &
weather2);
        void update(Weather day);
        friend istream operator >>(istream & ins, Weather & w);
      private:
        string date;
        int highTemp;
        int lowTemp;
        int rainfall;




};
#endif


//4.2
//cpp file
#include <iostream>
#include <istream>
#include <cstring>
#include "Weather.h"
using namespace std;

Weather::Weather()
{
      highTemp = -99;
      lowTemp = +99;
      rainfall = 0;

}
Weather::Weather(string dat, int ht, int lt, int rf)
{
      date = dat;
      highTemp = ht;
      lowTemp = lt;
      rainfall = rf;

}
Weather::~Weather()
{
}
int Weather::getRainFall() const
{
```

```
	return rainfall;
}
bool operator >(const Weather& weather1, const Weather & weather2)
{
	if(weather1.rainfall > weather2.rainfall)
		return true;
	else
		return false;

}
void Weather::update(Weather day)
{


}

istream operator >>(istream & ins, Weather & w)
{
	ins >> w.date >> w.highTemp >> w.lowTemp >> w.rainfall;
	return ins;
}



//Main function
//4.3


 //1
#include "Weather.h"
//2

//3
monthW(monthNames[m],-99,99,0
//4
open("monthNames[m].dat")

//5
if(fileName.fail())
{
	cout << "Failed to open a file" << endl;
	exit(1);
}
//6
Weather day1
//7
day1 >>

//8
day.update(day1);
//9
fileName.close();
//10

//11

//12

//13

//14a)

//14b)
```

Question 5

```
//5.1
#include "Project.h"

class ProjectWithMarks : public Project
{
	public:
	  ProjectWithMarks(int pNumber, string pLecture, string
pDescription);
	  void addStudent(string pStdNumber, string pName, string pMarks);
	  void addMark(string stdnum, string stdnam);
	  double calcAverage();

	private:
	  vector <int> studentMarks;
}

//5.2



//5.3

void ProjectWithMarks::addMark(string stdnum, string stdnam)
{

}
//5.4


//Question 6

//6.1
template <class T>
class Database
{
	public:
	  Database();
	  void insert(T n);
	  void swap(T pos1, T pos2);
	  T count();
	private:
	  vector<T> myData;

}
//6.2
template <class T>
void Database::swap(T pos1, T pos2)
{
	pos1 = pos2;
	pos2 = pos1;
}

//6.3
```