Lab 8

1: read and run following code in your computer and see the result (I won’t check this)

```
#include<iostream>
#include<string>
#include<cstdlib>
using namespace std;
void keep() { char c; cin >> c; }
void main()
{
	cout << "the value of left: " << endl;
	int x = 3;
	int y = x++;
	cout <<"x=3:      " << x << endl<<endl;
	cout <<"y=x++:    "<< y<<endl<<endl;
	int z = ++x;
	cout <<"z=++x:    " << z<<endl<<endl;
	int *p = new int[4]{1,4,7,10};
	int a = *p;
	int b = *p++;// derefer p, then increase the address
	int b1 = *p;
	int c = ++*p;
	int d = *(++p);
	int e = *(p--);
	int f = *++p;
	int g = (*p)++;
	int g1 = *p;

	cout <<"a=*p:     "<< a << endl<<endl;
	cout <<"b=*p++:   "<<b << endl<<endl;
	cout <<"b1 = *p:  "<< b1 << endl<<endl;
	cout << "c=++*p:   " << c << endl << endl;
	cout <<"d=*(++p): "<<d<< endl<<endl;
	cout <<"e=*(p--): "<< e << endl<<endl;
	cout <<"f=*++p:   "<<f << endl<<endl;
	cout <<"g=(*p)++: "<< g << endl<<endl;
	cout <<"g1=*p:    "<<g1 << endl<<endl;

	keep();

}
```

2,

(a) Declare a struct with name: Name_pair , with member: string name, int age

(b) Implement a function: add_pair, it’ll use an uninitializied vector<Name_pair> as its parameter. This function will ask user to enter name and age and add them to this vector, and this input process will continue until user enter a string “exit”.

(c) Implement a function: sort, it’ll sort the element of vector created above with an order of age from young to old (or small to large order).