Template

Problem 1: Will the code below compile?

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

template<typename T>
T minimum(const T& a, const T& b)
{
    return (a < b) ? a : b;
}
int main()
{
    int x = minimum(3, 5);
    cout << "x is: "<< x << endl;
    double y = minimum(3.2, 5.2);
    cout << "y is: "<< y << endl;
    int z = minimum(3.2, 5.2);
    cout << "z is: "<< z << endl;
    int m = minimum(3, 4.7);
    return 0;
}
```

Problem 2: What is the output of the code?

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

template<typename T>
T minimum(const T& a, const T& b)
{
    cout << "Use template minimum()"<<endl;
    return (a < b) ? a : b;
}
double minimum(const double& a, const double& b){
    cout << "Use double minimum()"<<endl;
    return (a < b) ? a : b;
}

int main()
{
    int x = minimum(3, 5);
    cout << "x is: "<< x << endl;
    double y = minimum(3.2, 5.2);
    cout << "y is: "<< y << endl;
    return 0;
}
```

Problem 3: Will the code below compile?

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

class Animal
{
public:
    Animal(int weight):m_weight(weight){};
    int getWeight() const {return m_weight;};
private:
    int m_weight;
};

template<typename T>
bool lessThan(const T& a, const T& b)
{
    return (a < b);
}

int main()
{
    Animal a(10);
    Animal b(15);
    cout << lessThan(a, b) <<endl;
    return 0;
}
```

Problem 4: Complete the class definition

```
_____________________________
class HoldOneValue {
public:
    void setValue(T value){_____________________};
    T getValue();
private:
    T m_value;
};


_____________________________
_________________::getValue()
{
    return m_value;
}
```

STL(Standard Template Library)

How to print all elements of vector<int> or list<int>?

```
vector<int> li;
for (_____________________________________________)
{
    cout<<___________<<endl;
}

list<int> li;
for (_____________________________________________)
{
    cout<<___________<<endl;
}
```

What's the problem with the code below?

```
for (vector<int>::iterator it = li.begin(); it != li.end(); it++)
{
    if(*it > 2)
        li.erase(it);
}
```