Template and STL

What's the problem with the code?

```
class Animal
{
public:
    Animal(int weight):m_weight(weight){};
    int getWeight() const {return m_weight;};
    list<int> getData() const{return data;};
    void printData() const{
        for (list<int>::const_iterator it = data.begin();
             it != data.end();it++)
        {
            cout << *it << endl;
        }
        cout << "END" <<endl;
    };
private:
    int m_weight;
    list<int> data;
};
template<typename T>
bool lessThan(const T& a,const T& b)
{
    return (a < b );
}
int main()
{
    Animal a(10);
    a.printData();
    a.getData().push_back(3);
    a.printData();
    return 0;
}
```

Recursion

Problem 1: Print the elements of an array in order

```
void printArrayInOrder(int arr[], int n)
{









}
```

How to print the array in reverse order?

Problem 2: Tower of Hanoi

```
//Only one disk can be moved at a time.
//Each move consists of taking the upper disk from one of the stacks
and placing it on top of another stack i.e. a disk can only be moved
if it is the uppermost disk on a stack.
//No disk may be placed on top of a smaller disk.

void solveHanoi(int n, int src, int dest, int buf)
{









}
```

Problem 3: Parade organization

```
//You are asked to organize a parade consisting of bands and floats.
//You can't place a band immediately after another.
//How many ways can you organize a parade of size n?
//i.e. for n = 2, you have 3 ways to organize the parade: float-float,
float-band and band-float.

void solveParade(int n)
{









}
```

Problem 4: How many ways are there to choose k out of n things? (k<=n)

```
int choose(int k, int n)
{









}
```

Inheritance & Polymorphism

(From http://netlab.cs.ucla.edu/~schoi/cs32/notes/cs32w11mid.pdf)

```
class A
{
public:
    A():m_msg("Apple"){}
    A(string msg):m_msg(msg){}
    virtual ~A(){ message();}
    void message()const { cout<<m_msg<<endl;}
private:
    string m_msg;
};

class B:public A
{
public:
    B() :A("Orange") {}
    B(string msg) : A(msg), m_a(msg){}
    void message() const { m_a.message();}
private:
    A m_a;
};
int main(int argc, const char * argv[]) {
    A *b1 = new B;
    B *b2 = new B;
    A *b3 = new B("Apple");
    b1->message();
    b2->message();
    b3->message();
    delete b1;
    delete b2;
    delete b3;
    return 0;
}
```

What is the output of the program?

Now make A's message() virtual, i.e. virtual void message() const;
What is the output of the program?

Stack & Queue
Using a stack, Check if the parentheses in a expression are balanced.
For example:
((2 + 4) * (15 – 20)) balanced
((12 + 30))) not balanced

```
#include<stack>
using namespace std;
bool balanced(const string &exp)
{









}
```