```cpp
//11 题答案
//时间复杂度最好为 O(1),最差为 O(listSize)
template<class T>
void arrayList<T>::push_back(const T&x)
{
    if(listSize==arrayLength)
    {
        changeLength(element,listSize,listSize*2);
        arrayLength*=2;
    }

    element[listSize++]=x;
}

//12 题答案
//时间复杂度为 O(1)
template<class T>
void arrayList<T>::pop_back()
{
    if(listSize<=0)
    {
        cout<<"already empty"<<endl;
        return;
    }
    else
     element[--listSize].~T();
}


//14 题答案
//时间复杂度为 O(1)
template<class T>
void arrayList<T>::reserve(int theCapacity)
{
    if(theCapacity<=0)
    {
        cout<<"theCapacity should be >0";
        return;
    }
    arrayLength=max(arrayLength,theCapacity);
}

//16 题答案
//时间复杂度为 O(1)
```

```
template<class T>
void arrayList<T>::clear()
{
    listSize=0;
    arrayLength=0;
    delete []element;
    element=NULL;
}
```

测试

```
#include"array_list.hpp"
int main()
{
    arrayList<int>s(1);
    s.push_back(33);
    cout<<s<<endl;
    s.push_back(2);
    cout<<s<<endl;
    cout<<"capacity:"<<s.capacity()<<endl;
    s.push_back(3);
    cout<<s<<endl;
    cout<<"capacity:"<<s.capacity()<<endl;
    s.pop_back();
    cout<<s<<endl;
    s.reserve(20);
    cout<<"capacity:"<<s.capacity()<<endl;
    s.clear();
    cout<<"capacity:"<<s.capacity()<<endl;
}
```

输出

```
33
33 2
capacity:2
33 2 3
capacity:4
33 2
capacity:20
capacity:0
```