

1. ArrayList

时间限制：3s，空间限制：256MB

题目描述

Rainy Memory 嫌弃 C++ 的数组很久了：不仅不会检查下标越界，也没有数组拼接、数组输出等方便的功能。学习了 OOP 知识的你，可不可以写一个类帮他解决这个问题呢？

你需要完成一个模板类 `ArrayList`，有且仅有一个模板参数 `T`，为数组中元素的类型。

这个类需要支持以下功能：

- 构造函数
 - 从已有的数组构造：参数 `T* arr, int length`，`length` 为 `arr` 的长度
 - 构造一个固定长度的数组：参数 `int length`，其中的元素应调用其默认构造函数
 - 拷贝构造函数（深拷贝，需要复制元素的值）
- 析构函数
- 成员函数 `size()`，返回数组大小。
- `operator[]` 的重载，支持通过 `[]` 赋值与得到数组元素（需要同时完成左值与右值的重载）。
- `operator+` 的重载，拼接两个数组，原来的两个数组不应该改变。
- `operator=` 的重载，将本数组赋值为另一个数组，需要支持连续赋值。（同拷贝构造函数一样是深拷贝）
- `operator==, !=` 的重载，数组相同的判定条件是长度与对应位元素均相等。
- `operator<<` 的重载，格式为使用一个空格隔开数组中的所有元素，不输出换行符。
 - 在评测时，程序的任何输出都将通过调用 `printList` 函数来实现（见下方代码，略去了模板相关的内容）。因此，如果你没有完成 `operator<<` 的重载，你也可以利用给定的 `print` 函数并注释掉 `#define USE_OS` 以使得评测能够正常进行，否则可能产生编译错误。`print` 函数同样按照格式输出到 `cout`，不输出换行符。

```
void printList(ArrayList list) {
#ifdef USE_OS
    std::cout << list << "\n";
#else
    list.print();
    std::cout << "\n";
#endif
}
```

保证模板参数 `T` 有默认构造函数，支持 `operator==, <<, =`。

同时需要在出现问题时抛出异常。我们提供以下异常类：

```

class error : public std::exception {
private:
    std::string msg;

public:
    explicit error(const char *_msg_) : msg(_msg_) {}

    const char *toString() {
        return msg.c_str();
    }
};

```

可能出现的异常有：

- 下标越界：下标为 0-based，若下标越界抛出 `index out of bound`
- 初始化长度非法：若构造函数中 `length < 0`，抛出 `invalid length`

调用示例

```

#include "ArrayList.hpp"
...
try {
    ArrayList<char> list_err(-1);
} catch (error &e) {
    std::cout << e.toString() << std::endl;
}

ArrayList<int> list(5);

try {
    list[-1];
} catch (error &e) {
    std::cout << e.toString() << std::endl;
}
try {
    list[2000];
} catch (error &e) {
    std::cout << e.toString() << std::endl;
}

for (int i = 0; i < list.size(); i++) list[i] = i;
printList(list);
ArrayList<int> temp = list;
printList(temp);
temp = list;
printList(temp);
if (temp == list) std::cout << "true" << std::endl;

int arr[5] = {4, 215, 12, 84, 76};
const ArrayList<int> list_const(arr, 5);
std::cout << list_const[3] << std::endl;
printList(list_const);

printList(list_const + list);

return 0;

```

...

样例输出

```
invalid length
index out of bound
index out of bound
0 1 2 3 4
0 1 2 3 4
0 1 2 3 4
84
4 215 12 84 76
4 215 12 84 76 0 1 2 721894 4
```

输入格式

你不需要处理输入，可以参考上述调用示例。

输出格式

你不需要处理输出，可以参考上述调用示例。

数据范围

check point	数据特征
1	同调用样例
2	仅检测异常处理
3	鲁棒性
4	主要考察+，size
5	流输出运算符重载
6	元素默认构造函数的调用
7 ~ 12	内容同上述1 ~ 6，检查内存泄漏，共20分