

# 复旦大学计算机科学技术学院

## 2021~2022 学年第二学期期末模拟考试答案

课程名称：\_\_\_\_面向对象程序设计\_\_\_\_

课程代码：\_\_\_\_COMP130135.03\_\_\_\_

开课院系：\_\_\_\_计算机科学技术学院\_\_\_\_

考试形式：线上考试（闭卷）

一、选择题（每题只选择一个答案；如选多个，不计分。每题 3 分，总分 24 分）

1、\_\_\_\_B\_\_\_\_      2、\_\_\_\_D\_\_\_\_      3、\_\_\_\_D\_\_\_\_      4、\_\_\_\_B\_\_\_\_

5、\_\_\_\_D\_\_\_\_      6、\_\_\_\_A\_\_\_\_      7、\_\_\_\_D\_\_\_\_      8、\_\_\_\_C\_\_\_\_

二、程序阅读题（每题 5 分，共 20 分）

1、4 0

2、12

3、8

4、Man display.  
Human over.

三、程序填空题（每空 3 分，共 24 分）

① T& it1, T& it2

② begin != end

③ minv = it = begin

（装订线内不要答题）

#### 四、编程题（2 题，共 32 分）

```
//Integer.h
#ifndef GUARD_INTEGER_H
#define GUARD_INTEGER_H

#include <iostream>
#include <vector>

class Integer {
    friend std::istream& operator>>(std::istream&, Integer&);
    friend Integer operator+(const Integer&, const Integer&);
    friend Integer operator*(const Integer&, const Integer&);
public:
    typedef std::vector<int>::size_type size_type;
    size_type size()const { return data.size(); }
    int& operator[](size_type i) { return data[i]; }
    const int& operator[](size_type i)const { return data[i]; }
private:
    std::vector<int> data;
    void carry(); //进位
};

std::ostream& operator<<(std::ostream&, const Integer&);

#endif

//Integer.cpp
#include <string>
#include "Integer.h"

using std::istream;    using std::ostream;
using std::string;

istream& operator>>(istream& is, Integer& Int)
{
    string s;
    is >> s; //正向读入
    for (Integer::size_type i = s.size(); i != 0; --i) //反向保存
        Int.data.push_back(s[i-1] - '0');
    return is;
}

ostream& operator<<(ostream& os, const Integer& Int)
{
    for (Integer::size_type i = Int.size(); i != 0; --i) //反向输出
        os << Int[i - 1];
    return os;
}

void Integer::carry()
{

```

```
int cc = 0;
for (Integer::size_type i = 0; i != size(); ++i) {
    data[i] += cc;
    cc = data[i] / 10;
    data[i] %= 10;
}
while (cc) {
    data.push_back(cc % 10);
    cc /= 10;
}
}

Integer operator+(const Integer& Int1, const Integer& Int2)
{
    Integer Int;
    Integer::size_type i1 = 0, i2 = 0;
    while (i1 != Int1.size() || i2 != Int2.size()) {
        int sum = 0;
        if (i1 != Int1.size()) {
            sum += Int1[i1]; i1++;
        }
        if (i2 != Int2.size()) {
            sum += Int2[i2]; i2++;
        }
        Int.data.push_back(sum);
    }
    Int.carry();

    return Int;
}

Integer operator*(const Integer& Int1, const Integer& Int2)
{
    Integer Int;
    Integer::size_type i1, i2;
    for (i2 = 0; i2 != Int2.size(); ++i2)
        for (i1 = 0; i1 != Int1.size(); ++i1) {
            if (i1 + i2 + 1 > Int.size()) Int.data.push_back(0);
            Int[i1 + i2] += Int1[i1] * Int2[i2];
        }
    Int.carry();

    return Int;
}
```