

编程作业: 期末考试—编程试题

您还未提交解答。您必须获得 80/100 分才能通过。

截止时间 在以下日期前通过此作业 六月 11, 11:59 晚上 PDT

说明

我提交的作业

讨论

编程题 # 1: 输出200

来源: 北京大学在线程序评测系统POJ (Coursera声明: 在POJ上完成的习题将不会计入 Coursera的最后成绩。)

总时间限制: 1000ms 内存限制: 1024kB

描述

使以下代码输出结果为200.

```
1  #include<iostream>
2  using namespace std;
3  class Number {
4  public:
5      int num;
6      Number(int n=0): num(n) {}
7  // 在此处补充你的代码
8  };
9
10 int main() {
11     Number n1(10), n2(20);
12     Number n3;n3 = n1*n2;
13     cout << int(n3) << endl;
14     return 0;
15 }
```

输入

不需要输入。

输出

输出结果为200。

样例输入

1 不需要输入。

样例输出

1 200

编程题 # 2：输出指定结果一

来源: 北京大学在线程序评测系统POJ (Coursera声明：在POJ上完成的习题将不会计入Coursera的最后成绩。)

总时间限制: 1000ms 内存限制: 1024kB

描述

填写代码，使输出结果为

2

2

8

10

```
1 #include <iostream>
2 using namespace std;
3 class Number {
4 public:
5     int num;
6     Number(int n): num(n) {
7     }
8     // 在此处补充你的代码
9 };
10 int main() {
11     Number a(2);
12     Number b = a;
13     cout << a.value() << endl;
14     cout << b.value() << endl;
15     a.value() = 8;
16     cout << a.value() << endl;
17     a+b;
18     cout << a.value() << endl;
19     return 0;
20 }
```

输入

不需要输入。

输出

使输出结果为

2

2

8

10

样例输入

1 不需要输入。

样例输出

```
1 2
2 2
3 8
4 10
```

编程题 # 3：计算数列平方和

来源: 北京大学在线程序评测系统POJ (Coursera声明：在POJ上完成的习题将不会计入Coursera的最后成绩。)

总时间限制: 1000ms 内存限制: 1024kB

描述

请写出sum函数，使其可以计算输入数列的平方和。

```
1  #include <iostream>
2  using namespace std;
3  // 在此处补充你的代码
4  int sqr(int n) {
5      return n * n;
6  }
7  int main() {
8      int t, n, a[0x100];
9      cin >> t;
10     for (int c = 0; c < t; ++c) {
11         cin >> n;
12         for (int i = 0; i < n; ++i) cin >> a[i];
13         cout << sum(a, n, sqr) << endl;
14     }
15     return 0;
16 }
```

输入

第一行是一个整数 t ($t \leq 10$), 表示数据组数;

每组输入数据包含两行, 第一行是一个整数 n ($n \leq 100$),

第二行是 n 个用空格分隔开的整数

输出

对每组输入数据, 输出该组数据中 n 个整数的平方和

样例输入

```
1  2
2  2
3  4 3
4  3
5  0 1 2
```

样例输出

```
1 25
2 5
```

编程题 # 4: 计算整数平方和

来源: 北京大学在线程序评测系统POJ (Coursera声明: 在POJ上完成的习题将不会计入Coursera的最后成绩。)

总时间限制: 1000ms 内存限制: 1024kB

描述

下列程序每次读入一个整数N, 若N为0则退出, 否则输出N和N的平方。

```
1  #include <iostream>
2  using namespace std;
3  // 在此处补充你的代码
4  int main(int argc, char* argv[]) {
5      CType obj;
6      int n;
7      cin>>n;
8      while ( n ) {
9          obj.setvalue(n);
10         cout<<obj++<<" "<<obj<<endl;
11         cin>>n;
12     }
13     return 0;
14 }
```

输入

K个整数。除最后一个数据外, 其他数据均不为0。

输出

K-1行。第I行输出第I个输入数和它的平方。

样例输入

```
1 1 5 8 9 0
```

样例输出

```
1 1 1
2 5 25
3 8 64
4 9 81
```

编程题 # 5：计算数组的低3位之和

来源: 北京大学在线程序评测系统POJ (Coursera声明：在POJ上完成的习题将不会计入Coursera的最后成绩。)

总时间限制: 1000ms 内存限制: 1024kB

描述

输入一个正整数构成的数组a[0], a[1], a[2], ..., a[n-1], 计算它们的二进制低3位之和。

```
1 #include <iostream>
2 #include <vector>
3 #include <algorithm>
4 using namespace std;
5 // 在此处补充你的代码
6 int main(int argc, char* argv[]) {
7     int v, my_sum=0;
8     vector<int> vec;
9     cin>>v;
10    while ( v ) {
11        vec.push_back(v);
12        cin>>v;
13    }
14    for_each(vec.begin(), vec.end(), CMy_add(my_sum
15        ));
16    cout<<my_sum<<endl;
17    return 0;
18 }
```

输入

数组a,以0表示输入结束。

输出

一个整数，所输入数组各元素的二进制低3位之和。

样例输入

```
1 1 3 9 7 3 6 20 15 18 17 4 8 18 0
```

样例输出

编程题 # 6: MyString

来源: 北京大学在线程序评测系统POJ (Coursera声明: 在POJ上完成的习题将不会计入Coursera的最后成绩。)

总时间限制: 1000ms 内存限制: 1024kB

描述

请写出 MyString类, 使得下面程序的输出结果符合下面的要求。

```

1  #include <cstring>
2  #include <cstdlib>
3  #include <string>
4  #include <iostream>
5  using namespace std;
6  // 在此处补充你的代码
7  int CompareString( const void * e1, const void * e2)
8  {
9      MyString * s1 = (MyString * ) e1;
10     MyString * s2 = (MyString * ) e2;
11     if( * s1 < *s2 )
12         return -1;
13     else if( *s1 == *s2)
14         return 0;
15     else if( *s1 > *s2 )
16         return 1;
17 }
18 int main()
19 {
20     MyString s1("abcd-"),s2,s3("efgh-"),s4(s1);
21     MyString SArray[4] = {"big","me","about","take"};
22     cout << "1. " << s1 << s2 << s3<< s4<< endl;
23     s4 = s3;
24     s3 = s1 + s3;
25     cout << "2. " << s1 << endl;
26     cout << "3. " << s2 << endl;
27     cout << "4. " << s3 << endl;
28     cout << "5. " << s4 << endl;
29     cout << "6. " << s1[2] << endl;
30     s2 = s1;
31     s1 = "ijkl-";
32     s1[2] = 'A' ;
33     cout << "7. " << s2 << endl;
34     cout << "8. " << s1 << endl;
35     s1 += "mnop";
36     cout << "9. " << s1 << endl;
37     s4 = "qrst-" + s2;
38     cout << "10. " << s4 << endl;
39     s1 = s2 + s4 + " uvw " + "xyz";
40     cout << "11. " << s1 << endl;
41     qsort(SArray,4,sizeof(MyString),CompareString);
42     for( int i = 0;i < 4;i ++ )
43         cout << SArray[i] << endl;
44     //s1的从下标0开始长度为4的子串
45     cout << s1(0,4) << endl;
46     //s1的从下标5开始长度为10的子串
47     cout << s1(5,10) << endl;
48     return 0;
49 }

```

输入

无

输出

1. abcd-efgh-abcd-

2. abcd-

3.

4. abcd-efgh-

5. efgh-

6. c

7. abcd-

8. ijAl-

9. ijAl-mnop

10. qrst-abcd-

11. abcd-qrst-abcd- uvw xyz

about

big

me

take

abcd

qrst-abcd-

样例输入

1 无

样例输出

```
1 1. abcd-efgh-abcd-
2 2. abcd-
3 3.
4 4. abcd-efgh-
5 5. efgh-
6 6. c
7 7. abcd-
8 8. ijAl-
9 9. ijAl-mnop
10 10. qrst-abcd-
11 11. abcd-qrst-abcd- uvw xyz
12 about
13 big
14 me
15 take
16 abcd
17 qrst-abcd-
```

编程题 # 7：字符串排序

来源: 北京大学在线程序评测系统POJ (Coursera声明: 在POJ上完成的习题将不会计入Coursera的最后成绩。)

总时间限制: 1000ms 内存限制: 1024kB

描述

请按照要求对输入的字符串进行排序。

```

1  #include <iostream>
2  #include <string>
3  #include <list>
4  using namespace std;
5
6  class A{
7  private:
8      string name;
9  public:
10     A(string n) :name(n){}
11     friend bool operator < (const class A& a1, const
        class A &a2);
12     friend bool operator == (const class A &a1, const
        class A &a2){
13         if (a1.name.size() == a2.name.size())
14             return true;
15         else
16             return false;
17     }
18     friend ostream & operator << (ostream &o, const A
        &a){
19         o << a.name;
20         return o;
21     }
22     string get_name() const{
23         return name;
24     }
25     int get_size() const{
26         return name.size();
27     }
28 };
29 // 在此处补充你的代码
30 int main(int argc, char* argv[])
31 {
32     list<A> lst;
33     int ncase, n, i = 1;
34     string s;
35     cin >> ncase;
36     while (ncase--){
37         cout << "Case: "<<i++ << endl;
38         cin >> n;
39         for (int i = 0; i < n; i++){
40             cin >> s;
41             lst.push_back(A(s));
42         }
43         lst.sort();
44         Show(lst.begin(), lst.end(), Print());
45
46         cout << endl;
47         lst.sort(MyLarge<A>());
48         Show(lst.begin(), lst.end(), Print());
49         cout << endl;
50         lst.clear();
51     }
52     return 0;
53 }

```

输入

第一行是正整数T，表示测试数据的组数

每组测试数据输入共两行，

第一行是正整数N，表示字符串个数

第二行是N个字符串, 字符串间用空格分离

输出

对于每组测试数据，先输出一行：

Case: n

如对第一组数据就输出Case: 1

第二行按照字符串长度从小到大排序之后输出N个字符串，字符串之间以空格间隔（不会出现字符串长度相同的情况）

第三行按照字符串首字符ASCII码序从小到大排序之后输出N个字符串，字符串之间以空格间隔（不会出现字符串首字母相同的情况）

样例输入

```
1 2
2 4
3 a bnss ds tsdfasg
4 5
5 aaa bbbb ccccd sa q
```

样例输出

```
1 Case: 1
2 a ds bnss tsdfasg
3 a bnss ds tsdfasg
4 Case: 2
5 q sa aaa bbbb ccccd
6 aaa bbbb ccccd q sa
```

编程题 # 8：计算整数k

来源: 北京大学在线程序评测系统POJ (Coursera声明：在POJ上完成的习题将不会计入Coursera的最后成绩。)

总时间限制: 1000ms 内存限制: 1024kB

描述

输入整数 n ($0 \leq n \leq 2^{30}$), 以及整数 i, j ($0 \leq i, j < 31, i < j-1$), 输出整数 k (按十六进制输出结果), k 的第 i 位和 n 相同, 第 j 位和 n 不同, i, j 之间的位是1, 其他位都是0。这里提到的所有的位, 指的都是二进制位, 最右边算第0位。

输入

第一行是整数 t，表示数据组数。

每组输入数据是一行，三个整数 n,i和j。

输出

对每组输入数据，按十六进制输出结果。

样例输入

```
1 2
2 23 3 5
3 7 0 2
```

样例输出

```
1 30
2 3
```

提示

本题请写出完整的程序。

编程题 # 9： 人群的排序和分类

来源: 北京大学在线程序评测系统POJ (Coursera声明：在POJ上完成的习题将不会计入Coursera的最后成绩。)

总时间限制: 1000ms 内存限制: 1024kB

描述

对人群按照输入的信息进行排序和分类。

```

1  #include <iostream>
2  #include <set>
3  #include <iterator>
4  #include <algorithm>
5  using namespace std;
6  // 在此处补充你的代码
7  int main()
8  {
9
10     int t;
11     cin >> t;
12     set<A*,Comp> ct;
13     while( t -- ) {
14         int n;
15         cin >> n;
16         ct.clear();
17         for( int i = 0; i < n; ++i) {
18             char c; int k;
19             cin >> c >> k;
20
21             if( c == 'A')
22                 ct.insert(new A(k));
23             else
24                 ct.insert(new B(k));
25         }
26         for_each(ct.begin(),ct.end(),Print);
27         cout << "*****" << endl;
28     }
29 }

```

输入

第一行是整数t,表明一共t组数据. $t < 20$

对每组数据:

第一行是整数n,表示下面一共有n行。 $0 < n < 100$

下面的每行代表一个人。每行以一个字母开头，代表该人所属的类别，然后跟着一个整数，代表年龄。字母只会是 'A'或'B'。整数范围0到100。数据保证年龄都不相同。

输出

对每组输入数据，将这些人按年龄从小到大输出。每个人先输出类别，再输出年龄。每组数据的末尾加一行 "*****"

样例输入

```

1  2
2  4
3  A 3
4  B 4
5  A 5
6  B 6
7  3
8  A 4
9  A 3
10 A 2

```

样例输出

```
1 A 3
2 B 4
3 A 5
4 B 6
5 ****
6 A 2
7 A 3
8 A 4
9 ****
```

编程题 # 10： 输出指定结果二

来源: 北京大学在线程序评测系统POJ (Coursera声明：在POJ上完成的习题将不会计入Coursera的最后成绩。)

总时间限制: 1000ms 内存限制: 1024kB

描述

通过填空使得程序输出的结果符合下面的要求。

```
1  #include <iostream>
2  #include <map>
3  using namespace std;
4  // 在此处补充你的代码
5  int A::count = 0;
6  void func(B b) { }
7  int main()
8  {
9      A a1(5),a2;
10     cout << A::count << endl;
11     B b1(4);
12     cout << A::count << endl;
13     func(b1);
14     cout << A::count << endl;
15     A * pa = new B(4);
16     cout << A::count << endl;
17     delete pa;
18     cout << A::count << endl;
19     return 0;
20 }
```

输入

不需要输入。

输出

使得程序的输出结果是：

2

3

B::~destructor

A::~destructor

3

4

B::~destructor

A::~destructor

3

B::~destructor

A::~destructor

A::~destructor

A::~destructor

样例输入

1 不需要输入。

样例输出

```
1 2
2 3
3 B::~destructor
4 A::~destructor
5 3
6 4
7 B::~destructor
8 A::~destructor
9 3
10 B::~destructor
11 A::~destructor
12 A::~destructor
13 A::~destructor
```


提示

int A::count = 0; 这个变量是用来记录一共有多少个类A及类A的派生类的对象的。

编程题 # 11：数据库内的学生信息

来源: 北京大学在线程序评测系统POJ (Coursera声明：在POJ上完成的习题将不会计入Coursera的最后成绩。)

总时间限制: 3000ms 内存限制: 20480kB

描述

程序填空，使得下面的程序,先输出

(Tom,80),(Tom,70),(Jone,90),(Jack,70),(Alice,100),

(Tom,78),(Tom,78),(Jone,90),(Jack,70),(Alice,100),

(70,Jack),(70,Tom),(80,Tom),(90,Jone),(100,Alice),

(70,Error),(70,Error),(80,Tom),(90,Jone),(100,Alice),

然后，再根据输入数据按要求产生输出数据

```

1  #include <iostream>
2  #include <string>
3  #include <map>
4  #include <iterator>
5  #include <algorithm>
6  using namespace std;
7  // 在此处补充你的代码
8  struct Student
9  {
10     string name;
11     int score;
12 };
13 template <class T>
14 void Print(T first,T last) {
15     for(;first!= last; ++ first)
16         cout << * first << ", ";
17     cout << endl;
18 }
19 int main()
20 {
21
22     Student s[] = { {"Tom",80}, {"Jack",70},
23                     {"Jone",90},
24                     {"Tom",70}, {"Alice",100} };
25     MyMultimap<string,int> mp;
26     for(int i = 0; i<5; ++ i)
27         mp.insert(make_pair(s[i].name,s[i].score
28                             ));
29     Print(mp.begin(),mp.end()); //按姓名从大到小输出
30
31     mp.Set("Tom",78);
32     //把所有名为"Tom"的学生的成绩都设置为78
33     Print(mp.begin(),mp.end());
34
35     MyMultimap<int,string,less<int> > mp2;
36     for(int i = 0; i<5; ++ i)
37         mp2.insert(make_pair(s[i].score,s[i].name
38                             ));
39
40     Print(mp2.begin(),mp2.end());
41     //按成绩从小到大输出
42     mp2.Set(70,"Error");
43     //把所有成绩为70的学生，名字都改为"Error"
44     Print(mp2.begin(),mp2.end());
45     cout << "*****" << endl;
46
47     mp.clear();
48
49     string name;
50     string cmd;
51     int score;
52     while(cin >> cmd ) {
53         if( cmd == "A" ) {
54             cin >> name >> score;
55             if(mp.find(name) != mp.end() ) {
56                 cout << "erroe" << endl;
57             }
58             mp.insert(make_pair(name,score));
59         }
60         else if(cmd == "Q") {
61             cin >> name;
62             MyMultimap<string,int>::iterator
63                 p = mp.find(name);
64             if( p!= mp.end()) {
65                 cout << p->second << endl
66                 ;
67             }
68         }
69     }
70 }

```

```

60         }
61         else {
62             cout << "Not Found" <<
               endl;
63         }
64     }
65 }
66 return 0;
67 }

```

输入

输入数据的每一行，格式为以下之一：

A name score

Q name score

name是个不带个空格的字符串，长度小于 20

score是个整数，能用int表示

A name score 表示往数据库中新增一个姓名为name的学生，其分数为score。开始时数据库中一个学生也没有。

Q name 表示在数据库中查询姓名为name的学生的分数

数据保证学生不重名。

输入数据少于200,000行。

输出

对于每个查询，输出学生的分数。如果查不到，则输出 "Not Found"

样例输入

```

1  A Tom1 30
2  A Tom2 40
3  Q Tom3
4  A Tom4 89
5  Q Tom1
6  Q Tom2

```

样例输出

```
1 (Tom,80),(Tom,70),(Jone,90),(Jack,70),(Alice,100),
2 (Tom,78),(Tom,78),(Jone,90),(Jack,70),(Alice,100),
3 (70,Jack),(70,Tom),(80,Tom),(90,Jone),(100,Alice),
4 (70,Error),(70,Error),(80,Tom),(90,Jone),(100,Alice),
5 *****
6 Not Found
7 30
8 40
```

提示

1) 编写模板的时候，连续的两个 ">" 最好要用空格分开，以免被编译器看作是 ">>" 运算符。VS可能无此问题，但是Dev C++和服务器上的编译环境会有这个问题。

比如 `vector<vector<int>>` 有可能出错，要改成 `vector<vector<int> >`

2) 在模板中写迭代器时，最好在前面加上 `typename` 关键字，否则可能会编译错。VS可能无此问题，但是Dev C++和服务器上的编译环境会有这个问题。

How to submit

When you're ready to submit, you can upload files for each part of the assignment on the "My submission" tab.

