

# 任务一

---

## 问题描述

创建一个类Person作为基类，成员变量包括姓名name.性别sex.用户名username.密码password和年龄age,并创建派生类Student和Worker,其中Student类的成员变量还包括专业major、年级grade、绩点gpa, Worker类包括国籍nationality和薪水salary。 其他要求如下:

- 每个类都包含完整的显式构造函数、析构函数和拷贝构造函数。
- 基类中包含一个虚函数Printinfo(), 作用是输出该实例的全部成员变量，输出的格式参考下方输出示例。要求实现该虚函数在继承关系中的完整多态。
- 为派生类Student重载 << 流运算符，实现通过std:cout << student;可直接输出I'm a student named XXX的功能，其中XXX为该student的成员变量name
- 在各个类的构造析构中输出Class Constructed!,析构函数中输出Class Destructed!, 拷贝构造函数中输出Class Copied!,其中Class填写所在类的类名，例如Person Constructed!。通过查看运行输出结果，简单叙述构造，拷贝构造，析构函数被调用的时机以及它们的实际功能(叙述内容写在程序最后即可)。
- 实现系统的登录与注册,通过系统注册实现向系统内增添用户,注册时,name只可使用纯中文,password要求包含数字,字母,特殊符号。登录后,用户可通过Query函数执行查询操作,输出该username对应的所有信息。如果查询的学号不存在,应输出“Query no such ID",同时,worker类成员可通过Query函数实现对所有student类的信息的查看。
- 支持用户修改密码，且密码不可与之前密码重复

#### 主函数示例:

```
int main() {
    Person person("Allen", "male", 29);
    Student student("Frank", "male", 15, "Robotics", 17, 3.5);
    // 调用多态函数, 按要求输出信息
    Worker worker("Ben", "male", 45, "China", 3500.0f);
    person.PrintInfo();
    student.PrintInfo();
    worker.PrintInfo();
}
```

```
// 验证运算符重载的正确性
std::cout << student << std::endl;
// 对三个实例进行拷贝
Person person2 = person;
Student student2 = student;
Worker worker2 = worker;
// 对其中一个拷贝得到的实例调用多态函数, 输出信息
worker2.PrintInfo();
return 0;
}
```

#### 输出示例:

关闭构造、拷贝构造和析构函数中的输出时:

```
[Person]姓名: Allen, 性别: male, 年龄: 29
[Student]姓名: Frank, 性别: male, 年龄: 15, 专业: Robotics, 年级: 17, 绩点: 3.5
[Worker]姓名: Ben, 性别: male, 年龄: 45, 国籍: China, 薪水: 3500
I'm a student named Frank
[Worker]姓名: Ben, 性别: male, 年龄: 45, 国籍: China, 薪水: 3500
```

打开构造、拷贝构造和析构函数中的输出时:

```
Person Constructed!
Person Constructed!
Student Constructed!
Person Constructed!
Worker Constructed!
[Person]姓名: Allen, 性别: male, 年龄: 29
[Student]姓名: Frank, 性别: male, 年龄: 15, 专业: Robotics, 年级: 17, 绩点: 3.5
[Worker]姓名: Ben, 性别: male, 年龄: 45, 国籍: China, 薪水: 3500
I'm a student named Frank Person Copied!
Person Copied!
Student Copied!
Person Copied!
Worker Copied!
[Worker]姓名: Ben, 性别: male, 年龄: 45, 国籍: China, 薪水: 3500
Worker Destructed!
Person Destructed!
Student Destructed!
```

文件】

编程题 第1/3题

C++

0701819

3题

## 1、丢失报文的位置

某通信系统持续向外发送报文，使用数组nums保存n个最近发送的报文，用于在报文未达到对端的情况下重发。报文使用序号sn表示，序号sn按照报文发送顺序从小到大排序，相邻报文sn不完全连续且有可能相同。报文使用循环覆盖的方式保存，即nums数组填满后，从头开始保存新的报文。假设需要重发序号为sn的报文，请找出序号为sn的报文在数组中的开始位置和结束位置。

### 解答要求

时间限制: C/C++ 1000ms, 其他语言: 2000ms

内存限制: C/C++ 256MB, 其他语言: 512MB

### 输入

第一行输入: 数组nums的大小n, 取值范围[0,10000]

第二行输入: 数组中的所有报文的序号sn, sn取值范围[0,100000]。

第三行输入: 需要重发的报文序号sn, 取值范围[0,100000]

### 输出

start end

说明: start和end代表需要重发的报文序号sn在数组中的起始下标和结束下标

下一题

存题

```
1 #inc
2 #incl
3 using
4
5 int E
6 in
7 in
8 wh
9
10
11
12
13
14
15 }
16 e
17
18 }
19 }
20 return
21 }
22 pair<int, i
23 int stan
24 if (
25
26 }
```

一题

题目 1

com/iexam/1366212223726481409/examContent?examId=13768413188354

shixizhi.huawei.com正在共享您的屏幕。

停止共享

第2题

牛]

01819

3题

### 编程题 第1/3题

C++

第一行输入数组中的所有报文的序号sn，sn取值范围[0,100000]。

第三行输入:需要重发的报文序号sn，取值范围[0,100000]

#### 输出

start end

说明: start和end代表需要重发的报文序号sn在数组中的起始下标和结束下标

#### 样例1

复制 输入: 7

0 0 1 2 2 5 6  
1

复制 输出: 2 2

解释: nums数组大小为7

保存了7个报文, sn分别是0 0 1 2 2 5 6

sn为1的报文在数组中仅有1个, 下标是2, 因此输出2 2

#### 样例2

复制 输入: 7

0 0 1 2 2 5 6  
2

复制 输出: 3 4

```
1 #include<i
2 #include<v
3 using name
4
5 int ErFen(v
6 int lef
7 int rig
8 while (
9 int
10 if (
11
12 }
13 else
14
15 }
16 else
17 r
18 }
19 }
20 return -1;
21 }
22 pair<int, int>
23 int start =
24 if (sta
25 ret
26 }
```

题 未答 1

下一题

存疑

shixizhi.huawei.com正在共享您的屏幕。

停止共享



件】

编程题 第1/3题

C++

701819

复制 输出: 2 2

解释: nums数组大小为7  
保存了7个报文, sn分别是0 0 1 2 2 5 6  
sn为1的报文在数组中仅有1个, 下标是2, 因此输出2 2

样例2

复制 输入: 7

0 0 1 2 2 5 6  
2

复制 输出: 3 4

解释: nums数组大小为7  
保存了7个报文, sn分别是0 0 1 2 2 5 6  
sn为2的报文在数组中有2个, 下标分别是3, 4, 因此输出3 4

样例3

复制 输入: 7

4 4 7 8 2 3 4  
4

复制 输出: 6 1

解释: nums数组大小为7  
保存了7个报文, sn分别是4 4 7 8 2 3 4  
sn为4的报文在数组中有3个, 下标分别是0, 1, 6, 说明数组存在记录满了从头开始记录的情况, 输出6 1

```
1 #include<iostream>
2 #include<vector>
3 using namespace std;
4
5 int ErFen(vector<int> &nums, int left, int right) {
6     if (left > right) return -1;
7     while (left < right) {
8         int mid = (left + right) / 2;
9         if (nums[mid] < mid) left = mid + 1;
10        else right = mid;
11    }
12    return left;
13}
14
15 int main() {
16    int n;
17    while (cin >> n) {
18        vector<int> nums(n);
19        for (int i = 0; i < n; i++) {
20            int sn;
21            cin >> sn;
22            nums[i] = sn;
23        }
24        int start = 0, end = n - 1;
25        int result = -1;
26        while (start < end) {
27            int mid = (start + end) / 2;
28            if (nums[mid] < mid) start = mid + 1;
29            else end = mid;
30        }
31        result = start;
32    }
33    return result;
34}
```

一题

0 未答 1

下一题

存疑

shixizhi.huawei.com正在共享您的屏幕。

停止共享

hi.huawei.com/iexam/1366212223726481409/examContent?examId=13768413188354129948&sessionId=

件】

编程题 第1/3题

01819

3题

复制 输出: 3 4

解释: nums数组大小为7  
保存了7个报文, sn分别是0 0 1 2 2 5 6  
sn为2的报文在数组中有2个, 下标分别是3, 4, 因此输出3 4

样例3

复制 输入: 7  
4 4 7 8 2 3 4  
4

复制 输出: 6 1

解释: nums数组大小为7  
保存了7个报文, sn分别是4 4 7 8 2 3 4  
sn为4的报文在数组中有3个, 下标分别是0, 1, 6, 说明数组存在  
记录满了从头开始记录的情况, 输出6 1

样例4

复制 输入: 7  
4 4 7 8 2 3 4  
6

复制 输出: -1 -1

解释: nums数组大小为7  
保存了7个报文, sn分别是4 4 7 8 2 3 4

```
1 #include<ios
2 #include<vec
3 using namesp
4
5 int ErFen(vec
6     int left
7     int right
8     while (le
9         int m
10        if (m
11            re
12        )
13        else i
14            le
15        )
16        else {
17            ris
18        }
19    }
20    return -1;
21 }
22 pair<int, int> 2
23     int start =
24     if (star
25     return
26 }
```

shixizhi.huawei.com正在共享您的屏幕。 停止共享

第3题

编程题 第2/3题

## 2、快速传球

班级组织传球活动，男女同学随机排成 $m$ 行 $n$ 列队伍，第一列中的任意一个男同学都可以作为传球的起点，要求最终将球传到最后一列的任意一个男同学手里，求所有能够完成任务的传球路线中的最优路线（传球次数最少的路线）的传球次数。

传球规则：

1. 男同学只能将球传给男同学，不能传给女同学。
2. 球只能传给身边前后左右相邻的同学。
3. 如果游戏不能完成，返回-1。

说明：

1. 传球次数最少的路线为最优路线。
2. 最优路线可能不唯一，不同最优路线都为最少传球次数。

### 解答要求

时间限制: C/C++ 100ms, 其他语言: 200ms

内存限制: C/C++ 256MB, 其他语言: 512MB

### 输入

班级同学随机排成的 $m$ 行 $n$ 列队伍，1代表男同学，0代表女同学。

输入第一行包含两个用空格分开的整数 $m$  [1, 30]和 $n$  [1, 30]，表示 $m$ 行 $n$ 列的队伍；接下来 $m$ 行每行包含 $n$ 个用空格分开的整数 $val$

```
1
2 // we have
3 // If addit
4 #include<ios
5
6 using namespa
7 int main()
8 {
9     // please
10    int m;
11    int n;
12    m = 31;
13    n = 31;
14    int nums[m]
15    int set_i;
16    cin>>set_i;
17    int set_j;
18    cin>>set_j;
19    cout<<endl;
20    int nums2[set
21    int pj=0;
22
23    for(int i1=0;i1
24    {
25        nums2[i1]=
26    }
```

上一题

下一题

存疑

shixizhi.huawei.com正在共享您的屏幕。

停止共享

帮助



x +

wei.com/iexam/1366212223726481409/examContent?examId=1376841318835412994&sessionId=

### 编程题 第2/3题

题目名称

时间限制: C/C++ 100ms, 其他语言: 200ms

内存限制: C/C++ 256MB, 其他语言: 512MB

#### 输入

班级同学随机排成的m行n列队伍，1代表男同学，0代表女同学。  
输入第一行包含两个用空格分开的整数m [1, 30]和n [1, 30]，表示m行n列的队伍；接下来是m行，每行包含n个用空格分开的整数1或0。

#### 输出

最优路线的传球次数（最少传球次数）。

#### 样例1

复制 输入: 4 4

```
1 1 1 0
1 1 1 0
0 0 1 0
0 1 1 1
```

复制 输出: 3

解释:

```
1 1 1 0
1 1 1 0
0 0 1 0
0 1 1 1
```

上一题

下一题

存疑

未答 1

shixizhi.huawei.com正在共享您的屏幕。

停止共享

隐藏

C++

```
1
2 // we have d
3 // If additio
4 #include<iost
5
6 using namespac
7 int main()
8 {
9     // please d
10     int m;
11     int n;
12     m = 31;
13     n = 31;
14     int nums[m][n]
15     int set_i;
16     cin>>set_i;
17     int set_j;
18     cin>>set_j;
19     cout<<endl;
20     int nums2[set_i]
21     int pj=0;
22
23     for(int i1=0;i1<
24     {
25         nums2[i1] =0.
26     }
```



awei.com/iexam/1366212223726481409/examContent?examId=1376841318835412994&sessionId=

## 编程题 第2/3题

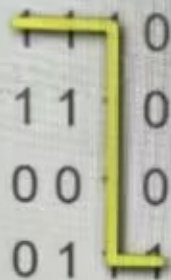
C++

9

0010  
0111

复制 输出: 5

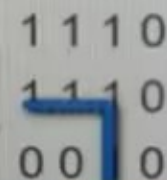
解释:



图一



图二



```
1
2 // we have d
3 // If additio
4 #include<iost
5
6 using namespac
7 int main()
8 {
9     // please d
10     int m;
11     int n;
12     m = 31;
13     n = 31;
14     int nums[m][n]
15     int set_i;
16     cin>>set_i;
17     int set_j;
18     cin>>set_j;
19     cout<<endl;
20     int nums2[set_i
21     int pj=0;
22
23     for(int i1=0;i1<
24     {
25         nums2[i1] =0;
26     }
```

上一题

下一题

存疑

|| shixizhi.huawei.com正在共享您的屏幕。

停止共享

全屏

未答 1

720p

× +

huawei.com/iexam/1366212223726481409/examContent?examId=1376841318835412994&sessionId=

编程题 第2/3题

19

3题

图一

1 1 1 0  
0 0 0  
0 1 1

图二

1 1 1 0  
1 1 1 0  
0 0 0  
0 1 1

图三

图一传球路线（黄色）需要传球6次。  
图二传球路线（红色）需要传球6次。  
图三传球路线（蓝色）需要传球5次，传球次数最少，为最优传球路线。

样例2

复制 输入: 3 4  
1 0 1 1  
1 1 0 0

上一题 下一题 存题

shixizhi.huawei.com正在共享您的屏幕。 停止共享 隐藏

```
1
2 // we have de
3 // If addition
4 #include<iostr
5
6 using namespace
7 int main()
8 {
9     // please d
10     int m;
11     int n;
12     m = 31;
13     n = 31;
14     int nums[m][n]
15     int set_i;
16     cin>>set_i;
17     int set_j;
18     cin>>set_j;
19     cout<<endl;
20     int nums2[set_i
21     int pj=0;
22
23     for(int i1=0;i1<
24     {
25         nums2[i1] =0;
```

第4题

编程题 第3/3题

319

3题

### 3、简易计算器

#### 题目描述

设计一款计算器软件，支持以下功能：

1) 支持let关键字

2) 支持通过let赋值表达式定义变量并初始化

例如：

```
let var1 = 123
```

```
let var=123
```

3) 变量需要先定义再引用，在表达式中引用未定义的变量，则表达式的结果也是未定义的

例如：

```
let var1 = 1
```

```
let var2 = var1 + 1 // var1是定义的
```

```
let var3 = var4 + 1 // var4是未定义的
```

```
let var4 = 1
```

4) 支持整数类型数据，整数数据的输入格式只需要支持十进制，支持负整数，整数取值范围 $-2147483648 \leq x \leq 2147483647$

例如：

```
let var3 = 10
```

```
let var3 = -10
```

5) 支持整数的加(+)、减(-)、乘(\*)、除(/)四则运算，四则运算符之间没

```
1
2 // we have
3 // IF addit
4 int main()
5 {
6     // please
7     // please
8     // please
9     return 0;
10 }
11
```

上一题

存疑

交卷

shixizhi.huawei.com正在共享您的屏幕。

停止共享



【

1819

3题

### 编程题 第3/3题

let var3 = 10

let var3 = -10

5) 支持整数的加(+)、减(-)、乘(\*)、除(/)四则运算。四则运算符之间没有优先级，运算数遵循左结合律，用例不考虑括号

例如：

let var4 = 1 + 2 \* var3

上述表达式的计算顺序是，先计算1+2结果为3，再将3乘以var3得到表达式的结果。

6) 支持通过out函数打印变量的值，函数参数只接受1个变量，不需要支持表达式

例如：

let var4 = 12

out(var4) // 将会输出12

7) 表达式中如果引用了未定义的变量，则表达式的结果是未定义的

8) 如果计算结果溢出，则表达式结果是溢出

9) 变量命名符合通用语言变量规范，必须是以下划线\_或者字母开头，遇到标点符号或者空格符时结束

例如：

let \_ = 1 // 变量名\_是合法的

let \_abc = 1 // 合法

let abc = 1 // 合法

let Abc\_1 = 1 // 合法

```
1
2 // we have
3 // IF addit
4 int main()
5 {
6     // pleas
7     // pleas
8     // pleas
9     return 0;
10 }
11
```

未答1

上一题

本题

交卷

shixizhi.huawei.com正在共享您的屏幕。

停止共享

退出



牛】

01819

3题

编程题 第3/3题

out(var4) // 将会输出12

- 7) 表达式中如果引用了未定义的变量，则表达式的结果是未定义的
- 8) 如果计算结果溢出，则表达式结果是溢出

- 9) 变量命名符合通用语言变量规范，必须是以下划线( )或者字母开头，遇到标点符号或者空格符时结束

例如：

let \_ = 1 // 变量名\_是合法的

let \_abc = 1 // 合法

let abc = 1 // 合法

let Abc\_1 = 1 // 合法

let abc.x = 1 // 非法

let abc.x = 1 // 非法

let 12abc = 1 // 非法

let abc x = 1 // 非法

解答要求

时间限制: C/C++ 1000ms, 其他语言: 2000ms

内存限制: C/C++ 32MB, 其他语言: 64MB

输入

输入描述

- 1) 每一行只有一个表达式

```
1
2 // we have
3 // IF addit
4 int main()
5 {
6     // pleas
7     // pleas
8     // pleas
9     return 0;
10 }
11
```

未答 1

上一题

存疑

交卷

shixizhi.huawei.com正在共享您的屏幕。

停止共享

隐藏

件]

01819

3题

### 编程题 第3/3题

时间限制: C/C++ 1000ms, 其他语言: 2000ms  
内存限制: C/C++ 32MB, 其他语言: 64MB

#### 输入

##### 输入描述

- 1) 每一行只有一个表达式
- 2) 最多支持24行输入
- 3) 每个用例输入至少有一个out输出表达式, 可以有多个out输出表达式
- 4) 每个变量只会赋值1次

例如:

```
let var1 = 1
let var2 = 3
let var3 = var1 + var2
out(var3)
```

#### 输出

##### 输出描述

- 1) 每遇到1个out输出表达式, 则打印输出变量的值
- 2) 对于out行, 只会输出一个out表达式的值
- 3) 如果out输出的变量未定义, 则打印<undefined>
- 4) 如果表达式结果发生了整数上溢或者下溢, 则对该变量的out输出表

C

```
1
2 // we have
3 // IF addit
4 int main()
5 {
6     // pleas
7     // pleas
8     // pleas
9     return 0;
10 }
11
```

未答 1

上一题

1/3

交卷

shixizhi.huawei.com正在共享您的屏幕。

停止共享

隐藏

件]

01819

3题

编程题 第3/3题

输出

输出描述

- 1) 每遇到1个out输出表达式，则打印输出变量的值
- 2) 对于out行，只会输出一个out表达式的值
- 3) 如果out输出的变量未定义，则打印<undefined>
- 4) 如果表达式结果发生了整数上溢或者下溢，则对该变量的out输出表达式输出<underflow>或者<overflow>
- 5) 如果表达式非法，则打印<syntax-error>

例如：

给定输入

```
let var1 = 1
let var2 = 3
let var3 = var1 + var2
out(var3)
out(var2)
out(var)
let var4 = -2147483649
let var5 = 2147483648
out(var4)
out(var5)
let x.y = 1
输出为
4
```

```
1
2 // we have d
3 // IF additio
4 int main()
5 {
6     // please
7     // please
8     // please
9     return 0;
10 }
11
```

上一题

存疑

交卷



【件】

701819

3题

编程题 第3题

下列代码在编译时，会报什么类型的错误？

例如：

给定输入

let var1 = 1

let var2 = 3

let var3 = var1 + var2

out(var3)

out(var2)

out(var)

let var4 = -2147483649

let var5 = 2147483648

out(var4)

out(var5)

let x.y = 1

输出为

4

3

<undefined>

<underflow>

<overflow>

<syntax-error>

样例1

C

```
1
2 // we have de
3 // IF additio
4 int main()
5 {
6     // please
7     // please
8     // please
9     return 0;
10 }
11
```

上一题

存疑

交卷



式页 x +

izhi.huawei.com/iexam/1366212223726481409/examContent?examId=13768413188354129948&sessionId=

【件】

701819

3题

编程题 第3/3题

out(var4)  
out(var5)  
let x.y = 1  
输出为

4  
3  
<undefined>  
<underflow>  
<overflow>  
<syntax-error>

样例1

复制 输入: let var1 = 1  
out(var1)

复制 输出: 1

解释:

样例2

复制 输入: out(var)

复制 输出: <undefined>

解释: 输出的变量var未定义

上一题 查看答案 交卷

```
1
2 // we have de
3 // IF addition
4 int main()
5 {
6     // please
7     // please
8     // please
9     return 0;
10 }
11
```

每题10分，请按照姓名+学号+任务一以.rar压缩格式发送至邮箱1036732248@qq.com