**RoboCup实验室2D及3D组2020年招新考试题**

# **姓名：** 付炎平

# **学号：** 2019217819

# **专业班级：** 19物联网二班

# 考试说明**!**

1. (附加题) 答对加分, 答错/不答 不扣分
2. **建议先写题目正文部分，情况调查部分可在提交考卷后再次提交补充**
3. **适当使用搜索引擎**，禁止搜索题目完整思路，但可以搜索如：*vector如何删除元素 等*问题
4. 如果发现题目有出错的地方欢迎指正 (有额外加分)
5. 完成后请在 **5月29日21:40** 前发送到邮箱 [hfutengine@163.com](mailto:hfutengine@163.com), 若因为时间关系没能在限定时间内完成所有题目（**包括附加题**），可以在完成后继续在第一次提交的基础上再次提交，我们也会酌情加分
6. 逾期提交需要在提交内容中说明, 否则将影响最终成绩

# 基本情况调查

（请根据实际情况作答，建议在**完成题目正文部分后再写！**）

个人情况调查：  
1.简单说说你目前了解过的一些算法（5~6个）并且曾用这些算法解决过什么问题？  
2.你都做过哪些项目？简单描述一下项目中所用到的技术。

实验室相关情况调查：  
1.请描述下对“课内学习以及课外实践孰轻孰重”这个问题的想法。

2.请描述你想要加入实验室的目的。

3.加入实验室后必定会占用你大量的时间，你对此有什么想法？

# 题目正文

（完成所有题目后可以考虑继续完成**附加题**）

**1. C++基础知识测试**

**1.1 STL的使用:**

**a.** 编写一个程序, 使用vector存储用户从键盘输入的n个整数, 利用STL中sort算法排序, 并用find方法查找某个数，最后删除尾部元素

#include<iostream>

#include<vector>

#include <algorithm>

using namespace std;

int main(){

int n=10;

int x;

vector<int> v;

for(int i=0;i<n;i++){

cin>>x;

v.push\_back(x);

}

sort(v.begin(),v.end());

find(v.begin(),v.end(),3);

v.pop\_back();

}

**b.** 使用set容器存储int类型元素, 编写函数求两个集合的交集.

#include<iostream>

#include<set>

#include <algorithm>

using namespace std;

int find(){

int a1[]={1,2,3,4,6};

int a2[]={2,3,4,5,8,10};

set<int>s1(a1,a1+5);

set<int>s2(a2,a2+6);

set<int>s3;

set<int>::iterator it1;

set<int>::iterator it2;

for(it1=s1.begin();it1!=s1.end();it1++)

for(it2=s2.begin();it2!=s2.end();it2++){

if(\*it1==\*it2){

s3.insert(\*it1);

}

}

set<int>::iterator it;

cout<<"俩集合的交集是：";

for(it=s3.begin();it!=s3.end();it++){

cout<<\*it<<" ";

}

}

int main(){

find();

}

**1.2 语言基础知识**

问题1. 变量 p 为指针变量，若 p=&a，下列表达式值为false的是（D ）。

A. &\*p==&a

B. \*&a==a

C. (\*p)++==a++

D. \*(p++)==++a

问题2. 字符串“RoboCup”存储在计算机中，需要的字节数是（B ）。

1. 7
2. 8
3. 14
4. 16

问题3. 什么是拷贝构造函数？深拷贝和浅拷贝的区别是什么？

浅拷贝是将原始对象中的数据型字段拷贝到新对象中去，将引用型字段的“引用”复制到新对象中去，不把“引用的对象”复制进去，所以原始对象和新对象引用同一对象，新对象中的引用型字段发生变化会导致原始对象中的对应字段也发生变化。

深拷贝是在引用方面不同，深拷贝就是创建一个新的和原始字段的内容相同的字段，是两个一样大的数据段，所以两者的引用是不同的，之后的新对象中的引用型字段发生改变，不会引起原始对象中的字段发生改变。

1. **代码调试能力测试**

请问下面这段代码存在的问题是什么？如何改正？

（提示：可以放到IDE中调试）

1. #include <iostream>
2. **using** std::cout;
3. **using** std::endl;
4. **class** Demo {
5. **public**:
6. **int**\* data;
7. **int** size;
8. Demo(**int** maxn) {
9. size = maxn;
10. data = **new** **int**[maxn];
11. }
12. ~Demo() {
13. **delete** data;
14. }
15. };
16. **void** print(Demo obj) {
17. cout << obj.data[0] << obj.data[1] << endl;
18. }
19. **int** main()
20. {
21. Demo obj{ 2 };
22. obj.data[0] = 1; obj.data[1] = 2;
23. print(obj);
24. }

21行应该是Demo obj(2);

16行print(Demo obj)函数时，由于Demo类没有实现复制构造函数，导致传参时没有把data里面的元素传进去，只需改成print(Demo &obj)；

1. **学习能力测试**

不知道大家有没有想过，为什么在初始化vector时，可以使用这样的方式初始化一个向量：

1. std::vector<**int**> vec{ 1, 2, 3, 4 };

可以发现我们在对vector进行初始化时，我们可以输入无限多的类型相同的参数，如{ 1, 2, 3, 4, 5, 6... }

大家如果学过Java等语言应该知道，在很多语言中函数的参数是可变的。这样对于函数参数个数不确定，而类型确定的情况而言，编程会变得更加方便和简洁。

在c++中也可以实现可变参数，但是会有些麻烦，不过c++11提供的**initializer\_list**可以帮助我们很方便地实现。

请你来实现一个数组类Array，并使用 c++11的 **initializer\_list** 重载自定义类Array的构造方法以实现可变参数的初始化该类。

数组类的基础模板（在此基础上添加内容）：

class Array

{

int\* data;

int size;

public:

Array();

~Array();

void show()

{

for (int i = 0; i < size; i++)

{

cout << data[i] << " ";

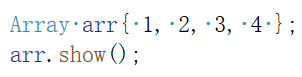
}

cout << endl;

}

};

要求：添加可变参数的构造函数，使得main函数中这样编写后，



输出为：



#include<iostream>

#include <algorithm>

using namespace std;

class Array

{

int\* data;

int size;

public:

Array(initializer\_list<int> list){

data=new int[list.size()];

size=list.size();

for(auto it = list.begin(),int i=0; it != list.end(); i++,++it){

data[i]=\*it;

}

}

~Array(){

delete []data;

}

void show()

{

for (int i = 0; i < size; i++)

{

cout << data[i] << " ";

}

cout << endl;

}

};

int main(){

Array array{1,2,3,4};

array.show();

}

1. **编程能力测试**

用户从键盘输入形式如 #ab!#12-45\*c\*#+\*f\* 这样的“数值范围”字符串，代表从12到45这个范围。12前面和45后面会有一些不包含数字以及英文连字符的干扰字符。 2. 请你编写一个类 Parse，可以解析这个字符串。然后提供两个函数，能够获取字符串中的第一个整数和第二个整数。

题目内容：

Parse类要提供一个有参构造函数，接收一个字符串参数；

Parse类要提供一个 int getFirst() 函数，返回“数值范围”字符串中的前面的整数；

Parse类要提供一个 int getLast() 函数，返回“数值范围”字符串中的后面的整数；

程序的主函数如下，请将主函数代码拷贝到你的开发环境中，但是不要修改主函数内部的代码：

1. #include <iostream>
2. #include <string>
3. **int** main() {
4. std::string s{};
5. std::cin >> s;  // 用户输入一个范围字符串
6. Parse p(s);     // 构造Parse对象p，同时解析字符串 s
7. std::cout << p.getFirst() << ' ' << p.getLast(); // 中间是两个单引号括起来的一个空格字符
8. **return** 0;
9. }

#include<iostream>

#include <string>

#include <math.h>

using namespace std;

class Parse{

public:

int a;

int b;

Parse(string s){

a=0;

b=0;

int s1[50];

for(int i=0;i<50;i++){

s1[i]=-1;

}

for(int i=0;i<s.length();i++){

if(s[i]>=48&&s[i]<=57){

s1[i]=s[i]-'0';

}

}

int j;

for(j=49;s1[j]==-1;j--){

}

int k;

for(k=0;s1[j-k]!=-1;k++){

a+=pow(10,k)\*s1[j-k];

}

int x;

for(x=j-k;s1[x]==-1;x--){

}

for(int p=0;s1[x-p]!=-1;p++){

b+=pow(10,p)\*s1[x-p];

}

}

int getFirst(){

return b;

}

int getLast(){

return a;

}

};

int main(){

string s="#ab!#129-450\*c\*#+\*f";

Parse P(s);

cout<<P.getFirst()<<endl<<P.getLast()<<endl;

}

****要求：****

1. 不要使用C语言的字符串处理函数

2. 使用string类自带的函数，或者标准库中的函数

提示：

1.在string中找'-'字符，可以使用string类中的find函数，或者find\_first\_of()函数。find\_first\_of()的说明：

<https://zh.cppreference.com/w/cpp/string/basic_string/find_first_of>

2.当然，你也可以用 find\_last\_of()或者rfind()函数。find\_last\_of()以及rfind()的说明：<https://zh.cppreference.com/w/cpp/string/basic_string/find_last_of>  
<https://zh.cppreference.com/w/cpp/string/basic_string/rfind>

3.将字符串的一部分提取出来，需要使用 substr() 成员函数。substr()的说明：<https://zh.cppreference.com/w/cpp/string/basic_string/substr>

4.将字符串转换为整数，可以使用标准库的 std::stoi() 函数。std::stoi的说明：

<https://zh.cppreference.com/w/cpp/string/basic_string/stol>

输入格式:

一个字符串，其中包含一个“数值范围”子串。该“数值范围”字串是用英文连字符分隔的两个整数构成的字符串，类似“321-54103”。整数前可能有字符0，例如“012-000”

该“数值范围”子串的****前面和后面可能会有0个或者多个干扰字符****。这些干扰字符不包含数字以及英文连字符。比如，干扰字符可能是 abc!@#$%^&\*()=+}{[]等

重要的事情说3遍：

****该“数值范围”子串的前面和后面可能会有0个或者多个干扰字符****

****该“数值范围”子串的前面和后面可能会有0个或者多个干扰字符****

****该“数值范围”子串的前面和后面可能会有0个或者多个干扰字符****

****输出格式：****

两个整数，中间用1个空格分隔。第二个整数后面没有空格或者换行

****输入样例：****

ab@r$&+fe321-54103@@@+==fsa.~

****输出样例：****

321 54103

题目来源：[https://www.icourse163.org/learn/BUPT-1003564002?tid=1450339466#/learn/ojhw?id=1222412217](https://www.icourse163.org/learn/BUPT-1003564002?tid=1450339466" \l "/learn/ojhw?id=1222412217)

可以在中国大学Mooc的OJ系统里测试程序的正确性。

**要求：给出较为全面的注释，解释你的代码。**

**附加题：**

1. 快速阅读该文章：<https://www.cnblogs.com/qicosmos/p/4325949.html>，寻找使用**可变模板参数**解决客观题中的第3题的方法，并加以实现。

提示：

* 使用逗号表达式展开参数包的方式可能会容易实现一些。
* 若要使用递归函数展开参数包，递归调用构造函数是不可行的（原因很明显）。