

CS150

初赛专题 集训



CS150 初赛专题集训公布资料的固定网站
请每次课前自行将资料下载到电脑

<https://pan.baidu.com/s/1XLQ1mwbfwfDTqFpzmn5HjA>


快快编程地址

<http://120.132.18.213:9062>

请登陆网站提交作业

集训做题步骤

- 1.在演草纸上填写答案
- 2.待所有空填写完毕后，打开程序，将草稿纸上的答案填入程序的空白处
- 3.运行程序，记录所有出现的错误
- 4.调试程序，尝试改正错误的答案
- 5.完成整个程序，可以顺利得到期望的结果
- 6.记录错误和心得经验



天下武功，唯快不破
完善程序，全靠猜测
大胆假设，小心求证
轻轻松松，蒙对几个

计数排序

```
#include<iostream>
#include<cstdio>
using namespace std;
int i,j,k,n,x,b[101];
int main() {
    cin>>n;
    for(i=0;i<=100;i++) b[i]=0;
    for(i=1;i<=n;i++) {
        cin>>x;
        b[x]=__ (1) __;
    }
    __ (2) __;
    for(i=0;i<=100;i++)
        while(__ (3) __) {
            cout<<__ (4) __;
            k=k+1;
            b[i]=b[i]-1;
            if (__ (5) __)
                cout<<endl;
        }
    cout<<endl;
    return 0;
}
```

输入n个0到100之间的整数，由小到大排序输出，每个数字以空格分隔，每行打印8个数字。

在演草纸上填写答案

计数排序

```
#include<iostream>
#include<cstdio>
using namespace std;
int i,j,k,n,x,b[101];
int main() {
    cin>>n;
    for(i=0;i<=100;i++) b[i]=0;
    for(i=1;i<=n;i++) {
        cin>>x;
        b[x]=__ (1) __;
    }
    __ (2) __;
    for(i=0;i<=100;i++)
        while(__ (3) __) {
            cout<<__ (4) __;
            k=k+1;
            b[i]=b[i]-1;
            if(__ (5) __)
                cout<<endl;
        }
    cout<<endl;
    return 0;
}
```

打开[计数排序.cpp](#)
填入答案

记录可能出现的错误

调试并完成程序，令
程序输出期望的结果


```

#include<iostream>
#include<cstdio>
using namespace std;
int i,j,k,n,x,b[101];
int main() {
    cin>>n;
    for(i=0;i<=100;i++) b[i]=0;
    for(i=1;i<=n;i++) {
        cin>>x;
        b[x]=__ (1) __;
    }
    __ (2) __;
    for(i=0;i<=100;i++)
        while(__ (3) __) {
            cout<<__ (4) __;
            k=k+1;
            b[i]=b[i]-1;
            if(__ (5) __)
                cout<<endl;
        }
    cout<<endl;
    return 0;
}

```

控制结构中出现k和b[i], k的作用是什么?
b数组干嘛用的? 为什么101个元素

程序的控制结构, 核心

k为什么+1

b[i]为什么-1?

打印的是什么?

格式控制, 每行打印8个数字,
换行

关键变量

```
#include<iostream>
#include<cstdio>
using namespace std;
int i,j,k,n,x,b[101];
int main() {
```

k表示总共输出了多少个数字
b计数器数组，统计数字x出现了多少次

```
    cin>>n;
    for(i=0;i<=100;i++) b[i]=0;
    for(i=1;i<=n;i++) {
        cin>>x;
        b[x]=__ (1) __;
    }
```

计数操作，变量的值b[x]++

```
    __ (2) __;
```

变量初始化

```
    for(i=0;i<=100;i++)
        while(__ (3) __) {
            cout<<__ (4) __;
            k=k+1;
            b[i]=b[i]-1;
            if(__ (5) __)
                cout<<endl;
        }
```

b[i]中还没全部打印

打印i

k是8的倍数换行

```
    cout<<endl;
    return 0;
```

```
}
```



```
#include<iostream>
using namespace std;
int i,j,k,n,x,b[101];
int main() {
    cin>>n;
    for(i=0;i<=100;i++) b[i]=0;
    for(i=1;i<=n;i++) {
        cin>>x;
        b[x]=b[x]+1;//(1)
    }
    k=0; //(2)
    for(i=0;i<=100;i++)
        while(b[i]>0) { //(3)
            cout<<i<<" "; //(4)
            k=k+1;
            b[i]=b[i]-1;
            if(k%8==0) //(5)
                cout<<endl;
        }
    cout<<endl;
    return 0;
}
```

找到关键变量

变量虽然很多，核心就那几个

一看变量是否定义。如果在程序中用到某些变量却没有定义，就要想到开始的空格是变量定义。

二看变量初始赋值。如果变量在程序中起作用，却没有设定初始值，要想到初始应该是多少。

三看变量值的变化。弄清变量的作用，例如计数器，累加器等等，要查明每次变化应该如何操作。

最后看变量的输入输出。在显示答案的地方，按照题目输出格式，检查每个变量是否都已打印。

判断质数

```
#include<iostream>
using namespace std;
int main() {
    int ____ (1) ____;
    cin>>n;
    if(n==2) cout<<____ (2) ____;
    else if(____ (3) ____ || n%2==0)
        cout<<"NO";
    else {
        i=3;
        while(i*i<=n) {
            if(____ (4) ____) {
                cout<<"NO";
                return 0;
            }
            i=i+2;
        }
        cout<<"YES";
    }
    return 0;
}
```

给出一个正整数，判断这个数是否是素数。

输入格式:

一个正整数

$n(1 \leq n \leq 10000)$

输出格式:

如果 n 是质数，输出YES，否则输出NO

输入样例:

10

输出样例:

NO

判断质数

```
#include<iostream>
using namespace std;
int main() {
    int ____ (1) ____;
    cin>>n;
    if(n==2) cout<<____ (2) ____;
    else if(____ (3) ____ || n%2==0)
        cout<<"NO";
    else {
        i=3;
        while(i*i<=n) {
            if(____ (4) ____) {
                cout<<"NO";
                return 0;
            }
            i=i+2;
        }
        cout<<"YES";
    }
    return 0;
}
```

程序中共出现了n和i变量，但是没有定义
n表示什么
i表示什么

```
#include<iostream>
using namespace std;
int main() {
    int n,i;  //(1)
    cin>>n;
    if(n==2) cout<<"YES";  //(2)
    else if(n==1 || n%2==0)  //(3)
        cout<<"NO";
    else {
        i=3;
        while(i*i<=n) {
            if(n%i==0) {  //(4)
                cout<<"NO";
                return 0;
            }
            i=i+2;
        }
        cout<<"YES";
    }
    return 0;
}
```

明确输入输出格式

程序虽然复杂，但输入输出是确定的

一、检查输入变量是否都已输入数值。如果题目要求的变量应当通过cin获取，但是还没有在程序中找到，那么空白处一定要输入变量的值。

二、检查题目要求的输出是否均已完成输出。如果题目要求的输出，还没有在程序中完整实现，那么要考虑到用cout打印结果。

三、输出要符合格式要求。例如换行，对齐，打印空格等等。

2013序列重排

```
#include<iostream>
using namespace std;
const int SIZE=10000;
int main() {
    int a[SIZE],b[SIZE];
    int i,j,n,p;
    cin>>n;
    for (i=1; i<=n; i++)
        cin>>a[i];
    cin>>__ (1) __;
    for (i=1; i<=p; i++)
        b[__ (2) __]=a[i];
    for (i=p+1; i<=n; i++)
        b[i-p] = ____ (3) ____;
    for (i=1; i<=____ (4) ____; i++)
        a[i]=b[i];
    for (i=1; i<=n; i++)
        cout<<a[i]<<" ";
    return 0;
}
```

将序列 a 的前 p 个数与后 $n-p$ 个数对调，且不改变这 p 个数（或 $n-p$ 个数）之间的相对位置。

例如，长度为5的序列1, 2, 3, 4, 5，当 $p=2$ 时重排结果为3, 4, 5, 1, 2。


```

#include<iostream>
using namespace std;
const int SIZE=10000;
int main() {
    int a[SIZE],b[SIZE];
    int i,j,n,p;
    cin>>n;
    for (i=1; i<=n; i++)
        cin>>a[i];
    cin>>__(1)__;
    for (i=1; i<=p; i++)
        b[__(2)__]=a[i];
    for (i=p+1; i<=n; i++)
        b[i-p] = __(3)__;
    for (i=1; i<=__(4)__; i++)
        a[i]=b[i];
    for (i=1; i<=n; i++)
        cout<<a[i]<<" ";
    return 0;
}

```

n,p由输入a得到

将a[]中的前p个数放到b[]中,
对应位置是哪里?

将a[]的后n-p个数放入到
b[]的对应位置

循环多少次?

}

```
#include<iostream>
using namespace std;
const int SIZE=10000;
int main() {
    int a[SIZE],b[SIZE];
    int i,j,n,p;
    cin>>n;
    for (i=1; i<=n; i++)
        cin>>a[i];
    cin>>p;  //(1)
    for (i=1; i<=p; i++)
        b[n-p+i]=a[i];  //(2)
    for (i=p+1; i<=n; i++)
        b[i-p] = a[i];  //(3)
    for (i=1; i<=n; i++)  //(4)
        a[i]=b[i];
    for (i=1; i<=n; i++)
        cout<<a[i]<<" ";
    return 0;
}
```

程序控制结构

万变不离其宗，从控制结构寻找突破

一、顺序结构。理清程序每一步要做的事情是什么？已经做了哪些，哪些还没有做，应当如何做？

二、选择结构。需要表达的条件是什么。明确条件表达式的求值，哪些情况取值true，哪些情况false。搞清楚多个条件之间的逻辑关系，与、或、非应当如何选？

三、循环结构。找出循环变量，从变量初始值，循环条件，循环变量自增三方面下手，分析循环的作用，查看哪些要素没有完全确定。

2015打印月历

输入月份 m ($1 \leq m \leq 12$)，按一定格式打印2018年第 m 月的月历

例如，2018年1月的月历打印效果如下（第一列为周日）

S	M	T	W	T	F	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

```

#include<iostream>
using namespace std;
const int dayNum[]={-1,31,28,31,30,31,30,31,31,30,31,30,31};
int m,offset,i;
int main() {
    cin>>m;
    cout<<"S\tM\tT\tW\tT\tF\tS"<<endl; //' \t'为TAB制表符
    ____ (1) ____;
    for (i=1; i<m; i++)
        offset = ____ (2) ____;
    for (i=0; i<offset; i++)
        cout<<' \t';
    for (i=1; i<=____ (3) ____; i++) {
        cout<<____ (4) ____;
        if (i==dayNum[m] || ____ (5) ____==0)
            cout<<endl;
        else
            cout<<' \t';
    }
    return 0;
}

```

```
#include<iostream>
using namespace std;
const int dayNum[]={-1,31,28,31,30,31,30,31,31,30,31,30,31};
int m,offset,i;
int main() {
    cin>>m;
    cout<<"S\tM\tT\tW\tT\tF\tS"<<endl; //'\t'为TAB制表符
    offset=1; //(1)
    for (i=1; i<m; i++)
        offset = (offset+dayNum[i])%7; //(2)
    for (i=0; i<offset; i++)
        cout<<'\\t';
    for (i=1; i<=dayNum[m]; i++) { //(3)
        cout<<i; //(4)
        if (i==dayNum[m] || (offset+i)%7==0) //(5)
            cout<<endl;
        else
            cout<<'\\t';
    }
    return 0;
}
```

2008字符串替换

给定一个字符串S（S仅包含大小写字母），下面的程序将S中的每个字母用规定的字母替换，并输出S经过替换后的结果。程序的输入是两个字符串，第一个字符串是给定的字符串S，第二个字符串S'由26个字母组成，它是a-z的任一排列，大小写不定。S'规定了每个字母对应的替换字母：S'中的第一个字母是字母A和a的替换字母，即S中的A用该字母大写替换，S中的a用该字母的小写替换，S'中的第二个字母是字母B和b的替换字母，即S中的B用该字母的大写，S中的b用该字母的小写替换，.....以此类推。


```

#include<iostream>
#include<cstring>
using namespace std;
char change[26],str[5000];
void CheckChangeRule() {
    int i;
    for(i=0;i<26;i++) {
        if(_____(1)_____)
            change[i]='A'-'a';
    }
}
void ChangeString() {
    int i;
    for(i=0;i<strlen(str);i++) {
        if(_____(2)_____)
            str[i]=change[str[i]-'A']-'a'+'A';
        else
            _____(3)_____;
    }
}
int main() {
    int i;
    cin>>str;
    cin>>change;
    CheckChangeRule();
    _____(4)_____;
    cout<<str<<endl;
    return 0;
}

```

```

#include<iostream>
#include<cstring>
using namespace std;
char change[26],str[5000];
void CheckChangeRule() {
    int i;
    for(i=0;i<26;i++) {
        if(change[i]>='A'&&change[i]<='Z') //(1)
            change[i]-='A'-'a';
    }
}
void ChangeString() {
    int i;
    for(i=0;i<strlen(str);i++) {
        if(str[i]>='A'&&str[i]<='Z') //(2)
            str[i]=change[str[i]-'A']-'a'+'A';
        else
            str[i]=change[str[i]-'a']; //(3)
    }
}
int main() {
    int i;
    cin>>str;
    cin>>change;
    CheckChangeRule();
    ChangeString(); //(4)
    cout<<str<<endl;
    return 0;
}

```

2014数字删除

下面程序的功能是将字符串中的数字字符删除后输出。

```
#include<iostream>
using namespace std;
int delnum(char *s) {
    int i,j;
    j=0;
    for(i=0; s[i]!='\0';i++)
        if(s[i]<'0'____(1)____s[i]>'9'){
            s[j]=s[i];
            ____ (2) ____;
        }
    return ____ (3) ____;
}
const int SIZE=30;
int main() {
    char s[SIZE];
    int len,i;
    cin.getline(s,sizeof(s));
    len=delnum(s);
    for(i=0;i<len;i++) cout<<____ (4) ____;
    cout<<endl;
    return 0;
}
```

```
#include<iostream>
using namespace std;
int delnum(char *s) {
    int i,j;
    j=0;
    for(i=0; s[i]!='\0';i++)
        if(s[i]<'0' || s[i]>'9'){ //(1)
            s[j]=s[i];
            j++; //(2)
        }
    return j; //(3)
}
const int SIZE=30;
int main() {
    char s[SIZE];
    int len,i;
    cin.getline(s,sizeof(s));
    len=delnum(s);
    for(i=0;i<len;i++) cout<<s[i]; //(4)
    cout<<endl;
    return 0;
}
```

2016读入整数

请完善下面的程序，使得程序能够读入两个int范围内的整数，并将这两个整数分别输出，每行一个。

输入的整数之间和前后只会出现空格或者回车。输入数据保证合法。

例如：

输入：

123 -789

输出：

123

-789

```
#include<iostream>
using namespace std;
int readint() {
    int num = 0;           // 存储读取到的整数
    int negative = 0;      // 负数标志
    char c;                // 存储当前读取到的字符
    c = cin.get();
    while ((c < '0' || c > '9') && c != '-')
        c = _____(1)_____;
    if (c == '-')
        negative = 1;
    else
        _____(2)_____;
    c = cin.get();
    while (_____(3)_____) {
        _____(4)_____;
        c = cin.get();
    }
    if (negative == 1)
        _____(5)_____;
    return num;
}
int main() {
    int a, b;
    a = readint();
    b = readint();
    cout << a << endl << b << endl;
    return 0;
}
```

```
#include<iostream>
using namespace std;
int readint() {
    int num = 0; // 存储读取到的整数
    int negative = 0;
    char c;
    c = cin.get();
    while ((c < '0' || c > '9') && c != '-')
        c = cin.get(); //(1)
    if (c == '-')
        negative = 1;
    else
        num = c - '0'; //(2)
    c = cin.get();
    while (c >= '0' && c <= '9') { //(3)
        num = num*10 + c - '0'; //(4)
        c = cin.get();
    }
    if (negative == 1)
        num = -num; //(5)
    return num;
}
int main() {
    int a, b;
    a = readint();
    b = readint();
    cout << a << endl << b << endl;
    return 0;
}
```


2004约瑟夫问题

原始的Joseph问题描述如下：有 n 个人围坐在一个圆桌周围，把这 n 个人依次编号为 $1, \dots, n$ ，从编号是1的人开始报数，数到第 m 个人出列，然后从出列的下一个入重新开始报数，数到第 m 个人又出列，.....如此反复，直到所有的人全部出列为止。比如当 $n=6, m=5$ 的时候，出列的顺序依次是5,4,6,2,3,1。现在的问题是：假设有 k 个好人和 k 个坏人。好人的编号从1到 k ，坏人的编号从 $k+1$ 到 $2k$ 。我们希望求出 m 的最小值，使得最先出列的 k 个人都是坏人。

输入格式：

一个正整数 k ($0 < k < 14$)

输出格式：

一个正整数 m

输入样例：

4

输出样例：

30

```

#include<iostream>
using namespace std;
long k,m,begin;
int check(long remain) {
    long result=(____(1)____)%remain;
    if(____(2)____) {
        begin=result; return 1;
    }
    else return 0;
}
int main() {
    long i,find=0;
    cin>>k;
    m=k;
    while(____(3)____) {
        find=1; begin=0;
        for(i=0;i<k;i++)
            if(!check(____(4)____)){
                find=0;break;
            }
        m++;
    }
    cout<<____(5)____;
    return 0;
}

```

```

#include<iostream>
using namespace std;
long k,m,begin;
int check(long remain) {
    long result=(begin+m-1)%remain;  //(1)
    if(result>=k) {  //(2)
        begin=result; return 1;
    }
    else return 0;
}
int main() {
    long i,find=0;
    cin>>k;
    m=k;
    while(!find) {  //(3)
        find=1; begin=0;
        for(i=0;i<k;i++)
            if(!check(2*k-i)){  //(4)
                find=0;break;
            }
        m++;
    }
    cout<<m-1;  //(5)
    return 0;
}

```

作业要求

- 1.在演草纸上填写答案
- 2.待所有空填写完毕后，补全程序，将草稿纸上的答案填入程序的空白处
- 3.运行程序，记录所有出现的错误
- 4.调试程序，尝试改正错误的答案
- 5.完成整个程序，可以顺利得到期望的结果
- 6.记录错误和心得经验