

青少年软件编程（C 语言）等级考试试卷（六级）

分数：100 题数：4

一、编程题(共 4 题，共 100 分)

1. 小白鼠再排队

N 只小白鼠($1 < N < 100$)，每只鼠头上戴着一顶有颜色的帽子。现在称出每只白鼠的重量，要求按照白鼠重量从小到大的顺序输出它们头上帽子的颜色。帽子的颜色用“red”，“blue”等字符串来表示。不同的小白鼠可以戴相同颜色的帽子。白鼠的重量用整数表示。

时间限制：1000

内存限制：65536

输入

输入第一行为一个整数 N，表示小白鼠的数目。 下面有 N 行，每行是一只白鼠的信息。第一个为正整数，表示白鼠的重量，不超过整型范围；第二个为字符串，表示白鼠的帽子颜色，字符串长度不超过 10 个字符。 注意：白鼠的重量各不相同。

输出

按照白鼠的重量从小到大的顺序输出白鼠的帽子颜色。

样例输入

```
3
30 red
50 blue
40 green
```

样例输出

```
red
green
blue
```

试题编号：20240323-6-01

试题类型：编程题

标准答案：

试题难度：一般

试题解析：

展示地址：[点击浏览](#)

考生答案：

```
#include <bits/stdc++.h>
```

```

using namespace std;
struct shu
{
    int n;
    string s;
}a[100010];
bool cmp(shu a,shu b)
{
    return a.n<b.n;
}
int main()
{
    int N;
    cin>>N;
    for(int i=0;i<N;i++)
    {
        cin>>a[i].n>>a[i].s;
    }
    sort(a,a+N,cmp);
    for(int i=0;i<N;i++)
    {
        cout<<a[i].s<<endl;
    }
}

```

考生得分：25

是否评分：已评分

评价描述：

2. 扩号匹配问题

在某个字符串（长度不超过 100）中有左括号、右括号和大小写字母；规定（与常见的算数式子一样）任何一个左括号都从内到外与在它右边且距离最近的右括号匹配。写一个程序，找到无法匹配的左括号和右括号，输出原来字符串，并在下一行标出不能匹配的括号。不能匹配的左括号用"\$"标注,不能匹配的右括号用"?"标注。

时间限制：3000

内存限制：65536

输入

输入包括多组数据，每组数据一行，包含一个字符串，只包含左右括号和大小写字母，字符串长度不超过 100

输出

对每组输出数据，输出两行，第一行包含原始输入字符，第二行由"\$","?"和空格组成，"\$"和"?"表示与之对应的左括号和右括号不能匹配。

样例输入

```
((ABCD(x)
)(rttyy())sss)(
```

样例输出

```
((ABCD(x)
$$
)(rttyy())sss)(
? ?$
```

试题编号：20240323-6-02

试题类型：编程题

标准答案：

试题难度：一般

试题解析：

展示地址：[点击浏览](#)

考生答案：

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
char a[101],b[101];
```

```
int up(int c)
```

```
{
    if(c==-1) return -1;
    else if(b[c]=='$') return c;
    else return up(c-1);
}
```

```
int main()
```

```
{
    while(scanf("%s",a)!=-1)
    {
        printf("%s\n",a);
        int n=strlen(a);
        memset(b,' ',sizeof(b));
        for(int i=0;i<n;i++)
        {
            if(a[i]=='(') b[i]=='$';
            else if(a[i]==')')
```

```

        {
            int m=up(i-1);
            if(m==-1) b[i]='?';
            else b[m]=' ';
        }
    }
    printf("%s\n",b);
}
return 0;
}

```

考生得分：25

是否评分：已评分

评价描述：

3. stack or queue

栈和队列都是常用的线性结构，它们都提供两个操作：

Push：加入一个元素。

Pop：弹出一个元素。

不同的是，栈是“先进后出”，而队列则是“先进先出”。

给出一个线性结构的进出顺序，判定这个结构是栈还是队列。

时间限制：1000

内存限制：65535

输入

第一行输入一个整数 t ，代表有 t 组测试数据 对于每组测试数据，第一行输入一个整数 n ，代表操作的次数。随后输入 n 行，每行包含两个整数 $type\ val$ 。当 $type = 1$ 时，表示该次操作为 push 操作， val 表示进入的数字。当 $type=2$ 时，表示该次操作为 pop 操作， val 代表出来的数字。 $3 \leq n \leq 2000$

输出

每组测试数据输出一行。输出该组数据对应的线性结构，“Stack” 或者 “Queue”。题目保证是栈或者队列的一种。

样例输入

```

2
6
1 1

```

```
1 2
1 3
2 3
2 2
2 1
4
1 1
1 2
2 1
2 2
```

样例输出

```
Stack
Queue
```

试题编号：20240323-6-03

试题类型：编程题

标准答案：

试题难度：一般

试题解析：

展示地址：[点击浏览](#)

考生答案：

```
#include <bits/stdc++.h>
```

```
using namespace std;
string Struct;
bool H;
int push[2005];
int pop[2005];
int main()
{
    int t;
    cin>>t;
    while(t--)
    {
        int n,type,val,lnum=0,Outnum=0;
        cin>>n;
        for(int i=0;i<n;++i)
```

```

{
    cin>>type>>val;
    if(type==1)
    {
        push[Innum++]=val;
    }
    else
    {
        pop[Outnum++]=val;
    }
}
int num=min(Innum,Outnum);
H=false;
for(int i=0;i<num;++i)
{
    if(push[i]!=pop[i])
    {
        Struct="Stack";
        H=true;
        break;
    }
}
if(!H) Struct="Queue";
cout<<Struct<<endl;
}
return 0;
}

```

考生得分：25

是否评分：已评分

评价描述：

4. 利用队列进行数字排序

对于 N 个数字，有人提出了如下的排序策略：

例如，对于数字 53、47、85、38、64、23

先建立 10 个队列(0 到 9)，用于存放数字的大小，将这 N 个数字依个位放入各自的队列之中，然后再按队列 0 到队列 9 依次出队。

例如，对于上面的数字，依次进队后，结果如下：

队列 3：53、23 队列 4：64 队列 5：85 队列 7：47 队列 8：38

将其依次出队后，结果为 53,23,64,85,47,38

然后，再将方才出队后的队对，依照十位放入各自的队列之中，然后再按队列 0 到队列 9 依次出队

例如，对于上面刚刚出队的序列 53,23,64,85,47,38，将其依次进队，结果如下：

队列 2：23 队列 3：38 队列 4：47 队列 5：53 队列 6：64 队列 8：85

将其依次出队后，结果为 23,38,47,53,64,85.因为这组数字最大只是两位数，所以排序结束。

如果还有更大的数字，那么，接下来就是其百位、千位……（如果位数不够，就补 0.比如最大的数字是四位数，那么数字 23 就当成 0023 处理）

请根据上述算法，对这些数字进行排序

时间限制：1000

内存限制：65535

输入

分为两行，第一行为一个数字 $N(1 \leq N \leq 1000)$ ，表示数字的个数 第二行为 N 个数字（都是非负数），以空格相隔，最大的数字不超过 9999

输出

输出两个部分 第一个部分为第一次进队出队的结果，先显示一行：Step1. 之后用 Queue0:...表示，共 10 行，结果用空格分隔，下同 之后为第二次进队出队的结果（如果需要第二次进队出队的话），先显示一行：Step2. 之后仍然用 Queue0:...表示，共 10 行 之后如果需要的话，则分别显示第三次、第四次的进队出队结果 第二部分为一行，即将数字排序后的结果（升序排序）

样例输入

20

```
41 67 34 0 69 24 78 58 62 64 5 45 81 27 61 91 95 42 27 36
```

样例输出

Step1.

Queue0:0

Queue1:41 81 61 91

Queue2:62 42

Queue3:

Queue4:34 24 64

Queue5:5 45 95

Queue6:36

Queue7:67 27 27

Queue8:78 58

Queue9:69

Step2.

Queue0:0 5

Queue1:

Queue2:24 27 27

Queue3:34 36

Queue4:41 42 45

Queue5:58

Queue6:61 62 64 67 69

Queue7:78

Queue8:81

Queue9:91 95

```
0 5 24 27 27 34 36 41 42 45 58 61 62 64 67 69 78 81 91 95
```

试题编号：20240323-6-04

试题类型：编程题

标准答案：

试题难度：一般

试题解析：

展示地址：[点击浏览](#)

考生答案：

```
#include<bits/stdc++.h>
```

```
using namespace std;
```

```
int get_digit_number(int x, int i)
{
    int tmp = pow(10, i);
    x /= tmp;
    return x % 10;
}
```

```
int how_many_digits(int x)
{
    int ret = 1;
    while ((x /= 10) != 0)
        ++ret;
    return ret;
}
```

```
vector<int> q[5][10];
```

```
int main()
{
    int n;
    int max_digits = 0;
    cin >> n;
    for (int i = 0; i < n; ++i)
    {
        int t; cin >> t;
        max_digits = max(max_digits, how_many_digits(t));
        q[0][t % 10].push_back(t);
    }
}
```

```
ostringstream sout;
for (int i = 1; i <= max_digits; ++i)
{
    printf("Step%d.\n", i);
    for (int j = 0; j < 10; ++j)
    {
        printf("Queue%d:", j);
        for (int e : q[i-1][j])
        {
            printf("%d ", e);
            if (i == max_digits)
                sout << e << " ";
            else
                sout << " ";
        }
    }
}
```

```
        q[i][get_digit_number(e, i)].push_back(e);
    }
    printf("\n");
}
}
cout << sout.str() << endl;
return 0;
}
```

考生得分：25

是否评分：已评分

评价描述：