

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. 字符串插入

有两个字符串 str 和 substr，str 的字符个数不超过 10，substr 的字符个数为 3。（字符个数不包括字符串结尾处的'\0'。）将 substr 插入到 str 中 ASCII 码最大的那个字符后面，若有多个最大则只考虑第一个。

时间限制：1000

内存限制：65536

输入

输入包括若干行，每一行为一组测试数据，格式为 str substr

输出

对于每一组测试数据，输出插入之后的字符串。

样例输入

```
abcab eee  
12343 555
```

样例输出

```
abceeeab  
12345553
```

试题编号：20230520-6-01

试题类型：编程题

标准答案：

试题难度：一般

试题解析：

```

#include<bits/stdc++.h>
using namespace std;
int main()
{
    string str,substr;
    while(cin>>str>>substr)
    {
        int maxn=0,maxid,i;
        for(i=0;i<str.size();i++)
        {
            if((str[i]-'0')>maxn)
            {
                maxn=str[i]-'0';
                maxid=i;
            }
        }
        for(i=0;i<str.size();i++)
        {
            cout<<str[i];
            if(i==maxid) cout<<substr;
        }
        cout<<endl;
    }
    return 0;
}

```

展示地址：[点击浏览](#)

考生答案：（此题已作答）

考生得分：25

是否评分：已评分

评价描述：

2. 机器翻译

VariantF 的电脑上安装了一个机器翻译软件，他经常用这个软件来翻译英语文章。这个翻译软件的原理很简单，它只是从头到尾，依次将每个英文单词用对应的中文含义来替换。对于每个英文单词，软件会先在内存中查找这个单词的中文含义，如果内存中有，软件就会用它进行翻译；如果内存中没有，软件就会在外存中的词典内查找，查出单词的中文含义然后翻译，并将这个单词和译义放入内存，以备后续的查找和翻译。

假设内存中有 M 个单元，每单元能存放一个单词和译义。每当软件将一个新单词存入内存前，如果当前内存中已存入的单词数不超过 M ，软件会将新单词存入一个未使用的内存单元；若内存中已存入 M 个单词，软件会清空最早进入内存的那个单词，腾出单元来，存放新单词。

假设一篇英语文章的长度为 N 个单词。给定这篇待译文章，翻译软件需要去外存查找多少次词典？假设在翻译开始前，内存中没有任何单词。

时间限制：1000

内存限制：262144

输入

第一行为两个正整数 M 和 N ，代表内存容量和文章的长度。第二行为 N 个非负整数，按照文章的顺序，每个数（大小不超过 1000000）代表一个英文单词。文章中两个单词是同一个单词，当且仅当它们对应的非负整数相同。对于 50% 的数据， $1 \leq N, M \leq 1000$ ；对于 100% 的数据， $1 \leq N, M \leq 1000000$ 。

输出

一个整数，为软件需要查词典的次数。

样例输入

```
3 7
1 2 1 5 4 4 1
```

样例输出

```
5
```

提示

整个查字典过程如下：每行表示一个单词的翻译，冒号前为本次翻译后的内存状况：空：内存初始状态为空。1. 1：查找单词 1 并调入内存。2. 1 2：查找单词 2 并调入内存。3. 1 2：在内存中找到单词 1。4. 1 2 5：查找单词 5 并调入内存。5. 2 5 4：查找单词 4 并调入内存替代单词 1。6. 2 5 4：在内存中找到单词 4。7. 5 4 1：查找单词 1 并调入内存替代单词 2。共计查了 5 次词典。

试题编号：20230520-6-02

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：（此题已作答）

考生得分：0

是否评分：已评分

评价描述：

3. 栈基本操作

依次读入序列元素 $1, 2, \dots, n$ 进栈，每进一个元素，机器可要求下一个元素进栈或弹栈，如此进行。给定一个输入序列，判断栈空时弹出的元素构成的序列是否可能等于给定的序列，如果是则输出栈的操作过程，否则输出“NO”。

时间限制：1000

内存限制：65535

输入

输入分两行 第一行为 n 的值（即序列元素个数） 第二行为给定的输入序列（序列元素均为整型）

输出

如果输入序列能够由题目规定的操作得到，则输出对栈的操作过程 否则直接输出“NO”

样例输入

```
7
4 5 3 6 2 7 1
```

样例输出

```
PUSH 1
PUSH 2
PUSH 3
PUSH 4
POP 4
PUSH 5
```

POP 5

POP 3

PUSH 6

POP 6

POP 2

PUSH 7

POP 7

POP 1

提示

给定序列中有可能有不在 $1\dots n$ 之间的数字

试题编号：20230520-6-03

试题类型：编程题

标准答案：

试题难度：一般

试题解析：

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
    int n;
    cin>>n;
    int a[100001],t=1,i;
    int l1[100001],l2[100001],l=0;
    stack <int> s;
    for(i=1;i<=n;i++)
        cin>>a[i];
    i=1;
    int f=0;
    while(1)
    {
        s.push(t);
        l++;
        l1[l]=0;
        l2[l]=t;
        t++;
        while(1)
        {
            if(s.size()==0) break;
            if(a[i]!=s.top()) break;
            i++;
        }
    }
}
```

```

    l++;
    l1[l]=1;
    l2[l]=s.top();
    s.pop();
}
if(s.size()==0 && t-1==n && i-1==n) break;
if(s.size()>0 && t-1==n && i-1<n)
{
    cout<<"NO";
    return 0;
}
}
for(i=1;i<=l;i++)
{
    if(l1[i]==1) cout<<"POP ";
    else cout<<"PUSH ";
    cout<<l2[i]<<endl;
}
return 0;
}

```

展示地址：[点击浏览](#)

考生答案：（此题已作答）

考生得分：25

是否评分：已评分

评价描述：

4. 双端队列

定义一个双端队列，进队操作与普通队列一样，从队尾进入。出队操作既可以从队头，也可以从队尾。编程实现这个数据结构。

时间限制：1000

内存限制：65535

输入

第一行输入一个整数 t ，代表测试数据的组数。 每组数据的第一行输入一个整数 n ，表示操作的次数。 接着输入 n 行，每行对应一个操作，首先输入一个整数 $type$ 。 当 $type=1$ ，进队操作，接着输入一个整数 x ，表示进入队列的元素。 当 $type=2$ ，出队操作，接着输入一个整数 c ， $c=0$ 代表从队头出队， $c=1$ 代表从队尾出队。 $n \leq 1000$

输出

对于每组测试数据，输出执行完所有的操作后队列中剩余的元素,元素之间用空格隔开，按队头到队尾的顺序输出，占一行。如果队列中已经没有任何的元素，输出 NULL。

样例输入

```
2
5
1 2
1 3
1 4
2 0
2 1
6
1 1
1 2
1 3
2 0
2 1
2 0
```

样例输出

```
3
NULL
```

试题编号：20230520-6-04

试题类型：编程题

标准答案：

试题难度：一般

试题解析：

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
    int t;
    cin>>t;
    while(t-->0)
    {
        int n;
        cin>>n;
        int a,b,i;
        int l[100001],head=1,tail=0;
        for(i=1;i<=n;i++)
```

```
{
    cin>>a>>b;
    if(a==1)
    {
        tail++;
        l[tail]=b;
    }
    else
    {
        if(b==0) head++;
        else tail--;
    }
}
if(tail-head<0 || head>tail) cout<<"NULL";
else
    for(i=head;i<=tail;i++)
        cout<<l[i]<<" ";
cout<<endl;
}
return 0;
}
```

展示地址：[点击浏览](#)

考生答案：（此题已作答）

考生得分：25

是否评分：已评分

评价描述：