

《数据结构与算法》第四周作业反馈

助教-陈孙一硕

A 线性表循环右移

循环右移 m 位，先输出最后 m 个数，再输出前 $n - m$ 个数即可。注意要先把 m 对 n 取模，否则会出错。

```
#include<bits/stdc++.h>
using namespace std;
#define int long long
const int N=2e6+101;
int a[N];
void solve(){
    int n,m; cin>>n>>m; m%=n;
    for (int i=1;i<=n;i++) cin>>a[i];
    for (int i=n-m+1;i<=n;i++) cout<<a[i]<<" ";
    for (int i=1;i<=n-m-1;i++) cout<<a[i]<<" ";
    cout<<a[n-m]<<endl;
}
signed main(){
    ios::sync_with_stdio(false);
    cin.tie(nullptr);
    cout.tie(nullptr);
    int T=1; //cin>>T;
    while (T--) solve();
}
```

B 顺序表区间数据求和

按题意模拟即可。

```

#include<bits/stdc++.h>
using namespace std;
#define int long long
const int N=2e6+101;
int a[N];
void solve(){
    int n,x,y; cin>>n;
    for (int i=1;i<=n;i++) cin>>a[i];
    cin>>x>>y;
    int ans=0;
    for (int i=1;i<=n;i++) if (a[i]>=x&&a[i]<=y) ans+=a[i];
    cout<<ans<<endl;
}
signed main(){
    ios::sync_with_stdio(false);
    cin.tie(nullptr);
    cout.tie(nullptr);
    int T=1; //cin>>T;
    while (T--) solve();
}

```

C 阶乘的递归实现

递归求解即可。

```

#include<bits/stdc++.h>
using namespace std;
#define int long long
int calc(int x){
    if (x==0) return 1;
    return x*calc(x-1);
}
void solve(){
    int n; cin>>n;
    cout<<calc(n)<<endl;
}

```

```

signed main(){
    ios::sync_with_stdio(false);
    cin.tie(nullptr);
    cout.tie(nullptr);
    int T=1; //cin>>T;
    while (T--) solve();
}

```

D 大整数加法

使用类与对象的写法固然可行，但是太过于繁琐。

按照字符串格式读入，按位进行存储。从低位到高位，依次相加，注意判断进位即可。

```

#include<bits/stdc++.h>
using namespace std;
#define int long long
const int N=2e6+101;
int a[N],b[N],ans[N];
string A,B;
void solve(){
    for (int i=0;i<=1000;i++) a[i]=b[i]=ans[i]=0;
    if (A.size()<B.size()) swap(A,B);
    for (int i=0;i<A.size();i++) a[i+1]=A[i]-'0';
    int tmp=A.size();
    for (int i=B.size()-1;i>=0;i--){
        b[tmp]=B[i]-'0';
        tmp--;
    }
    int plus=0;
    for (int i=A.size();i>=1;i--){
        ans[i]=(a[i]+b[i]+plus)%10;
        if (a[i]+b[i]+plus>=10) plus=1;
        else plus=0;
    }
    if (plus) cout<<1;
    for (int i=1;i<=A.size();i++) cout<<ans[i];
}

```

```

    cout<<endl;
}

signed main(){
    ios::sync_with_stdio(false);
    cin.tie(nullptr);
    cout.tie(nullptr);
    while (cin>>A>>B) solve();
}

```

E 出栈序列

按照题意模拟，如果到某一步，不论什么操作都进行不下去了，就判断不行。

```

#include <bits/stdc++.h>
using namespace std;
#define int long long
string in, out, ans = "";
int sz;
stack<char> stk;
bool flag = 1;
signed main() {
    ios::sync_with_stdio(false);
    cin.tie(nullptr);
    cin >> in >> out;
    sz = in.size();
    int i = 0, j = 0;
    while (1) {
        if (i == sz && j == sz) {
            break;
        }
        if (stk.empty()) {
            stk.push(in[i]);
            ans += 'P';
            i++;
        } else if (stk.top() == out[j]) {
            stk.pop();
            j++;
        } else {
            cout << "No" << endl;
            flag = 0;
            break;
        }
    }
    if (flag)
        cout << "Yes" << endl;
}

```

```

    ans += '0';
    j++;
} else {
    if (i == sz) {
        flag = 0;
        break;
    }
    stk.push(in[i]);
    ans += 'P';
    i++;
}
if (flag) {
    cout << "right" << endl;
    cout << ans << endl;
} else {
    cout << "wrong" << endl;
    string res = "";
    while (!stk.empty()) {
        res = stk.top() + res;
        stk.pop();
    }
    cout << res << endl;
}
return 0;
}

```