

[DSLK]. Bài 1. Insertion

Problem

Submissions

Leaderboard

Discussions

Cho danh sách liên kết đơn mỗi node trong DSLK lưu thông tin của 1 sinh viên gồm có : Mã sinh viên (chuỗi có không quá 20 kí tự), Họ tên (chuỗi không quá 50 kí tự) và điểm gpa. Có 3 thao tác sau : 1. Thêm 1 sinh viên vào đầu danh sách, 2. Thêm 1 sinh viên vào cuối danh sách, 3. Thêm 1 sinh viên vào vị trí K trong danh sách. Bạn hãy thực hiện in ra danh sách liên kết sau khi thực hiện 1 loạt thao tác ở trên.

Input Format

Dòng đầu tiên là N : số lượng thao tác cần thực hiện. Các dòng tiếp theo mô tả thao tác, mỗi thao tác gồm 4 dòng. Dòng 1 là loại thao tác(1, 2, 3), nếu thao tác là 3 thì số tiếp theo sau là vị trí chèn K, dòng 2 là mã sinh viên, dòng 3 là họ tên, dòng 4 là điểm gpa của sinh viên cần thêm vào DSLK. Đối với thao tác loại 3 khi kích thước của DSLK là N và chèn vào vị trí N + 1 tức là bạn thêm node đó vào cuối.

Constraints

$1 \leq N \leq 1000$; $0 \leq X \leq 1000$; $1 \leq K \leq \text{Size}(\text{DSLK}) + 1$

Output Format

In ra DSLK sau khi thực hiện N thao tác chèn, mỗi sinh viên in ra trên 1 dòng, các thông tin viết cách nhau một dấu cách và gpa lấy 2 số sau dấu phẩy.

Sample Input 0

```
7
1
CNTT1
Le Van Long
2.40
1
CNTT2
Le Van Long
1.20
1
CNTT3
Vuong Hoang Manh
1.20
1
CNTT4
Le Hoang Nam
2.80
2
CNTT5
Vuong Van Phuoc
2.60
1
CNTT6
Pham Xuan Manh
3.10
3 2
CNTT7
Le Xuan Phuoc
2.40
```

Sample Output 0

```
CNTT6 Pham Xuan Manh 3.10
CNTT7 Le Xuan Phuoc 2.40
CNTT4 Le Hoang Nam 2.80
CNTT3 Vuong Hoang Manh 1.20
CNTT2 Le Van Long 1.20
CNTT1 Le Van Long 2.40
CNTT5 Vuong Van Phuoc 2.60
```

[DSLK]. Bài 2. Xóa node

Problem

Submissions

Leaderboard

Discussions

Cho một DSLK, mỗi node trong DSLK là một số tự nhiên. Thực hiện xóa toàn bộ node có giá trị X trong DLKS. Ví dụ DSLK = {1, 2, 2, 3, 3, 2, 4} và X = 2 thì sau khi xóa DSLK sẽ là {1, 3, 3, 4}. Bài này các bạn phải cài đặt bằng DSLK.

Yêu cầu cài đặt hàm xóa node như sau :

```
void xoa(node *&head, int x){  
    //code  
}
```

Input Format

Dòng đầu tiên là N : số lượng node của DSLK và X là giá trị cần xóa. Dòng thứ 2 gồm N số lần lượt là các node trong DSLK.

Constraints

$1 \leq N \leq 1000$; Các node và X là số tự nhiên không quá 1000;

Output Format

In ra DSLK sau khi xóa hết các node có giá trị X. Nếu DSLK rỗng thì in ra EMPTY

Sample Input 0

```
8 7  
6 7 1 9 4 5 4 7
```

Sample Output 0

```
6 1 9 4 5 4
```

[DSLK]. Bài 3. Ngăn xếp

Problem

Submissions

Leaderboard

Discussions

Cài đặt cấu trúc dữ liệu ngăn lưu các số nguyên bằng DSLK. Ngăn xếp hỗ trợ 3 thao tác : Push, Pop, Show. Nếu thao tác là push bạn thêm 1 phần tử vào đỉnh ngăn xếp. Nếu thao tác là pop và ngăn xếp không rỗng thì bạn thực hiện xóa phần tử khỏi đỉnh ngăn xếp, nếu ngăn xếp rỗng thì không thực hiện xóa. Nếu thao tác là show thì bạn liệt kê các phần tử trong ngăn xếp theo thứ tự từ đỉnh trở xuống đáy ngăn xếp, trong trường hợp ngăn xếp rỗng thì in EMPTY và cách ra 1 dòng sau dòng EMPTY

Input Format

Dòng đầu tiên sẽ là N : số lượng thao tác. N dòng tiếp theo mô tả loại thao tác, nếu là push sẽ có thêm 1 số nguyên đi kèm.

Constraints

$1 \leq N \leq 1000$;

Output Format

In ra đáp án của bài toán

Sample Input 0

```
11
pop
show
pop
show
show
show
show
pop
push 119
push 433
pop
```

Sample Output 0

```
EMPTY
EMPTY
EMPTY
EMPTY
EMPTY
```

Sample Input 1

```
9
pop
push 928
push 619
show
show
pop
pop
show
push 761
```

Sample Output 1

```
619 928
619 928
EMPTY
```

[DSLK]. Bài 4. Hàng đợi

Problem

Submissions

Leaderboard

Discussions

Cho Hàng đợi lưu các số nguyên được cài đặt bằng DSLK, hàng đợi hỗ trợ 3 thao tác : Push, pop, front. Trong đó nếu thao tác là push thì thêm 1 phần tử vào cuối hàng đợi, thao tác là pop thì xóa phần tử khỏi đầu hàng đợi nếu hàng đợi không rỗng, front thì in ra đỉnh ở đầu hàng đợi nếu hàng đợi không rỗng, hàng đợi rỗng thì in ra EMPTY

Input Format

Dòng đầu tiên sẽ là N : số lượng thao tác. N dòng tiếp theo mô tả loại thao tác, nếu là push sẽ có thêm 1 số nguyên đi kèm.

Constraints

$1 \leq N \leq 1000$;

Output Format

In ra đáp án của bài toán

Output Format

In ra đáp án của bài toán

Sample Input 0

```
9
push 968
pop
push 851
push 51
front
pop
push 159
push 561
push 840
```

Sample Output 0

```
851
```

Sample Input 1

```
7
push 892
pop
pop
pop
pop
front
front
```

Sample Output 1

```
EMPTY
EMPTY
```

[DSLK]. Bài 5. Sắp xếp DSLK

Problem

Submissions

Leaderboard

Discussions

Cho DSLK các số nguyên, hãy thực hiện sắp xếp các node trong DSLK theo thứ tự tăng dần, giảm dần rồi in ra màn hình.

Input Format

Dòng đầu tiên là N : Số lượng node trong DSLK; Dòng thứ 2 gồm N số lần lượt là các node trong DSLK.

Constraints

$1 \leq N \leq 100$; Các node trong DSLK là số nguyên không âm không vượt quá 1000;

Output Format

Dòng 1 in ra DSLK được sắp xếp tăng dần; Dòng 2 in ra DSLK được sắp xếp giảm dần;

Sample Input 0

```
9
59 92 13 13 35 52 17 3 26
```

Sample Output 0

```
3 13 13 17 26 35 52 59 92
92 59 52 35 26 17 13 13 3
```


[DSLK]. Bài 6. Đếm tần suất

Problem

Submissions

Leaderboard

Discussions

Cho DSLK lưu các từ và tần suất của nó, nhiệm vụ của bạn là sử dụng DSLK để lưu các từ kèm theo tần suất xuất hiện của nó trong một đoạn văn.

Cấu trúc node sử dụng :

```
struct node{
    string data;
    int tansuat;
    node *next;
};
```

Hàm thêm từ vào DSLK :

```
void them(node *&head, string x){
    // code
}
```

Sử dụng hàm main có sẵn như sau :

```
int main(){
    node *head = NULL;
    string s;
    while(cin >> s){
        them(head, s);
    }
    duyet(head);
}
```

Input Format

Gồm nhiều dòng, mỗi dòng có thể có nhiều từ.

Constraints

Có tất cả không quá 10000 từ, mỗi từ có độ dài không quá 20 kí tự.

Output Format

In ra các từ kèm theo tần suất của nó theo thứ tự xuất hiện.

Output Format

In ra các từ kèm theo tần suất của nó theo thứ tự xuất hiện.

Sample Input 0

```
java  
python python backend      programming 28tech  
python
```

Sample Output 0

```
java 1  
python 3  
backend 1  
programming 1  
28tech 1
```

[DSLK]. Bài 7. Lật ngược DSLK đôi

Problem

Submissions

Leaderboard

Discussions

Cho DSLK đôi chứa các số nguyên, nhiệm vụ của bạn là lật ngược DSLK đôi này và in ra các node trong DSLK sau khi lật.

Sử dụng hàm main có sẵn như sau :

```
int main(){
    node *head = NULL;
    int n;
    cin >> n;
    while(n--){
        int x; cin >> x;
        themcuoi(head, x);
    }
    reverse(head);
    duyet(head);
}
```

Input Format

Dòng đầu tiên là N : số lượng node trong DSLK; Dòng thứ 2 gồm N số là các số trong DSLK;

Constraints

$1 \leq N \leq 1000$; Các node trong DSLK là số nguyên không âm không quá 1000.

Output Format

In ra DSLK đôi sau khi lật ngược.

Sample Input 0

```
9
97 207 886 278 987 602 826 817 459
```

Sample Output 0

```
459 817 826 602 987 278 886 207 97
```

[DSLK]. Bài 8. DSLK Đa thức

Problem

Submissions

Leaderboard

Discussions

Cho 2 đa thức $A(x)$ và $B(x)$, tiến hành tính tổng 2 đa thức này và in ra theo thứ tự bậc giảm dần. Sử dụng một node trong DSLK như sau :

```
4
5 □ struct node{
6     int bac;
7     int heso;
8     struct node *next;
9     };
10
```

Ví dụ $A(x) = 3x^2 + 5x + 7$, $B(x) = x^4 + 3x + 2$ thì bạn phải in ra đa thức tổng là $x^4 + 3x^2 + 3x + 9$.

Input Format

Dòng 1 là đa thức $A(x)$.

Dòng 2 là đa thức $B(x)$.

Constraints

Mỗi đa thức có bậc không quá 50, số lượng kí tự trong 1 đa thức không quá 10000 kí tự.

Output Format

In ra đa thức tổng của 2 đa thức A, B theo thứ tự giảm dần về bậc của biến.

Sample Input 0

```
1x^0 + 4x^3 + 5x^2 + 5x^1 + 1x^0 + 1x^1 + 3x^2 + 3x^3 + 1x^1
2x^0 + 5x^4 + 4x^0 + 4x^1 + 1x^0 + 2x^0 + 1x^3 + 3x^2 + 3x^3
```

Sample Output 0

```
5x^4 + 11x^3 + 11x^2 + 11x^1 + 11x^0
```

```

#include <bits/stdc++.h>

using namespace std;

struct node{
    int heso, somu;
    node *next;
};

node* makeNode(int heso, int somu){
    //cap phat dong
    node *ptr = new node;
    //gan du lieu
    ptr->heso = heso;
    ptr->somu = somu;
    ptr->next = NULL;
    return ptr;
}

void duyet(node *head){
    while(head != NULL){
        //truy cap du lieu cua node head
        cout << head->heso << "x^" << head->somu;
        //tu node hien => nhay ra node dung sau
        head = head->next; // i++
        if(head != NULL) cout << " + ";
    }
    cout << endl;
}

void them(node *&head, int hs, int sm){
    node *temp = head;
    node *prev = head;
    while(temp != NULL){
        if(temp->somu == sm){
            temp->heso += hs;
            return;
        }
        prev = temp;
        temp = temp->next;
    }
    node *newNode = makeNode(hs, sm);
    if(prev == NULL){
        head = newNode; return;
    }
    prev->next = newNode;
}

```

```

void sx(node *&head){
    for(node *i = head; i != NULL; i = i->next){
        node *min = i;
        for(node *j = i->next; j != NULL; j = j->next){
            if(min->somu < j->somu){
                min = j;
            }
        }
        swap(min->somu, i->somu);
        swap(min->heso, i->heso);
    }
}

int main(){
    node *head = NULL;
    string s;
    while(cin >> s){
        if(s != "+"){
            int hs = 0, sm = 0;
            int i = 0;
            while(s[i] != 'x'){
                hs = hs * 10 + (s[i] - '0');
                ++i;
            }
            i += 2;
            while(i < s.size()){
                sm = sm * 10 + (s[i] - '0');
                ++i;
            }
            them(head, hs, sm);
        }
    }
    sx(head);
    duyet(head);
}

```

[DSLK]. Bài 9. Liệt kê giá trị

Problem

Submissions

Leaderboard

Discussions

Cho một DSLK đôi chứa các số nguyên, nhiệm vụ của bạn là in ra các giá trị khác nhau trong DSLK theo thứ tự xuất hiện, mỗi giá trị chỉ liệt kê 1 lần

Input Format

Dòng đầu tiên là N : số lượng phần tử trong DSLK; Dòng thứ 2 gồm N số là các phần tử trong DSLK;

Constraints

$1 \leq N \leq 1000$; Các phần tử trong DSLK là số nguyên không âm 32 bit.

Output Format

In ra kết quả của bài toán

Sample Input 0

```
6
4 3 2 2 5 5
```

Sample Output 0

```
4 3 2 5
```


[DSLK]. Bài 10. GPA Sinh Viên

Problem

Submissions

Leaderboard

Discussions

Cho cấu trúc node để lưu trữ thông tin 1 sinh viên như sau :, (các bạn bổ sung phần tham chiếu vào nhé)

```
struct node{  
    char *id;  
    char *name;  
    double gpa;  
};
```

Tiến hành nhập danh sách sinh viên và tìm những sinh viên có điểm gpa cao nhất.

Input Format

Dòng đầu tiên là N : số lượng sinh viên; Các dòng tiếp là thông tin của sinh viên, mỗi thông tin của sinh viên gồm 3 dòng, dòng 1 là mã sinh viên, dòng 2 là tên, dòng 3 là gpa.

Constraints

$1 \leq N \leq 1000$; Tên và id là chuỗi không quá 100 ký tự, GPA là số thực nằm trong khoảng từ 0 tới 4.

Output Format

In ra những sinh viên có điểm cao nhất theo thứ tự xuất hiện

Sample Input 0

```
7
CNTT1
Vuong Duc Phuoc
2.80
CNTT2
Tran Xuan Phuoc
1.90
CNTT3
Pham Xuan Phuoc
2.90
CNTT4
Do Van Tuan
3.80
CNTT5
Vuong Hoang Hai
2.30
CNTT6
Tran Duc Phuoc
1.70
CNTT7
Pham Ngoc Phuoc
2.40
```

Sample Output 0

```
CNTT4 Do Van Tuan 3.80
```

```

#include <bits/stdc++.h>
using namespace std;

using ll = long long;

struct node{
    char *id; // id[100]
    char *name; // name[100]
    double gpa;
    struct node *next;
};

typedef struct node node;

double max_gpa = 0;

node *makeNode(){
    node *newNode = new node;
    newNode->id = new char[100];
    newNode->name = new char[100];
    cin.ignore();
    gets(newNode->id);
    gets(newNode->name);
    cin >> newNode->gpa;
    max_gpa = max(max_gpa, newNode->gpa);
    newNode->next = NULL;
    return newNode;
}

void pushBack(node *&head){
    node *newNode = makeNode();
    if(head == NULL){
        head = newNode;
        return;
    }
    node *temp = head;
    while(temp->next != NULL){
        temp = temp->next;
    }
    temp->next = newNode;
}

```

```
int main(){
    node *head = NULL;
    int n;
    cin >> n;
    while(n--){
        pushBack(head);
    }
    while(head != NULL){
        if(head->gpa == max_gpa){
            //printf("%s %s %.2lf\n", head->id, head->name, head->gpa);
            cout << head->id << ' ' << head->name << ' ' << fixed << setprecision(2) <<
head->gpa << endl;
        }
        head = head->next;
    }
}
```