**[DSLK]. Bài 1. Chèn**

* [**Vấn đề**](https://www.hackerrank.com/contests/28t222kech-lap-898inh-c-contest-12-danh-sach-lien-ket/challenges/dslk-bai-1-insertion)
* [**đệ trình**](https://www.hackerrank.com/contests/28t222kech-lap-898inh-c-contest-12-danh-sach-lien-ket/challenges/dslk-bai-1-insertion/submissions)
* [**bảng xếp hạng**](https://www.hackerrank.com/contests/28t222kech-lap-898inh-c-contest-12-danh-sach-lien-ket/challenges/dslk-bai-1-insertion/leaderboard)
* [**thảo luận**](https://www.hackerrank.com/contests/28t222kech-lap-898inh-c-contest-12-danh-sach-lien-ket/challenges/dslk-bai-1-insertion/forum)

Cho danh sách liên kết đơn từng nút trong DSLK lưu thông tin của 1 sinh viên bao gồm: Mã sinh viên (chuỗi không quá 20 ký tự), Họ tên (chuỗi không quá 50 ký 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 ra danh sách liên kết sau khi thực hiện 1 chuỗi thao tác ở trên.

**Định dạng đầu vào**

Dòng đầu tiên là N : số lượng thao tác cần thực hiện. Các dòng tiếp theo mô tả thao tác, mỗi thao tác bao gồm 4 dòng. Dòng 1 là loại thao tác(1, 2, 3), nếu thao tác là 3 thì số tiếp theo 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 nút đó vào cuối.

**Hạn chế**

1<=N<=1000; 0<=X<=1000; 1<=K<=Kích thước(DSLK) + 1

**Định dạng đầu ra**

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

**Đầu vào mẫu 0**

7

1

CNTT1

Lê Văn Long

2,40

1

CNTT2

Lê Văn Long

1,20

1

CNTT3 Vương Hoàng Mạnh

1,20 1 CNTT4 Lê Hoàng Nam 2,80

2 CNTT5 Vương Văn Phước 2,60 1 CNTT6 Phạm Xuân Mạnh 3,10 3 2 CNTT7 Lê Xuân Phước 2,40

**Đầu ra mẫu 0**

CNTT6 Phạm Xuân Mạnh 3,10

CNTT7 Lê Xuân Phước 2,40

CNTT4 Lê Hoàng Nam 2,80

CNTT3 Vương Hoàng Mạnh 1,20

CNTT2 Lê Văn Long 1,20

CNTT1 Lê Văn Long 2,40

CNTT5 Vương Văn Phước 2,60

**[DSLK]. Bài 2. Xóa nút**

* [**Vấn đề**](https://www.hackerrank.com/contests/28t222kech-lap-898inh-c-contest-12-danh-sach-lien-ket/challenges/dslk-bai-2-xoa-node)
* [**đệ trình**](https://www.hackerrank.com/contests/28t222kech-lap-898inh-c-contest-12-danh-sach-lien-ket/challenges/dslk-bai-2-xoa-node/submissions)
* [**bảng xếp hạng**](https://www.hackerrank.com/contests/28t222kech-lap-898inh-c-contest-12-danh-sach-lien-ket/challenges/dslk-bai-2-xoa-node/leaderboard)
* [**thảo luận**](https://www.hackerrank.com/contests/28t222kech-lap-898inh-c-contest-12-danh-sach-lien-ket/challenges/dslk-bai-2-xoa-node/forum)

Cho một DSLK, mỗi nút trong DSLK là một số tự nhiên. Thực hiện việc xóa toàn bộ nút 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.

**Định dạng đầu vào**

Dòng đầu tiên là N : số lượng nút của DSLK và X là giá trị cần xóa. Dòng thứ 2 bao gồm N number of the node in DSLK.

**Hạn chế**

1<=N<=1000; Các nút và X là số tự nhiên không quá 1000;

**Định dạng đầu ra**

In ra DSLK sau khi xóa hết các nút có giá trị X. If DSLK empty, in ra EMPTY

**Đầu vào mẫu 0**

8 7

6 7 1 9 4 5 4 7

**Đầu ra mẫu 0**

6 1 9 4 5 4

**[DSLK]. Bài 3. Ngăn xếp**

* [**Vấn đề**](https://www.hackerrank.com/contests/28t222kech-lap-898inh-c-contest-12-danh-sach-lien-ket/challenges/dslk-bai-3-ngan-xep)
* [**đệ trình**](https://www.hackerrank.com/contests/28t222kech-lap-898inh-c-contest-12-danh-sach-lien-ket/challenges/dslk-bai-3-ngan-xep/submissions)
* [**bảng xếp hạng**](https://www.hackerrank.com/contests/28t222kech-lap-898inh-c-contest-12-danh-sach-lien-ket/challenges/dslk-bai-3-ngan-xep/leaderboard)
* [**thảo luận**](https://www.hackerrank.com/contests/28t222kech-lap-898inh-c-contest-12-danh-sach-lien-ket/challenges/dslk-bai-3-ngan-xep/forum)

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 đẩy bạn thêm 1 phần tử vào ngăn xếp. Nếu thao tác là pop và Ngăn xếp không trống thì bạn thực hiện xóa phần tử khỏi ngăn xếp, nếu Ngăn xếp trống thì không thực hiện xóa. Nếu thao tác là hiển thị 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 trống thì in EMPTY và cách ra 1 dòng sau dòng EMPTY

**Định dạng đầu vào**

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

**Hạn chế**

1<=N<=1000;

**Định dạng đầu ra**

In ra câu trả lời của bài toán

**Đầu vào mẫu 0**

11

pop

show

pop

show

show

show

show

pop

push 119

push 433

pop

**Đầu ra mẫu 0**

TRỐNG

TRỐNG

TRỐNG

TRỐNG

TRỐNG

**Đầu vào mẫu 1**

9

pop

đẩy 928

đẩy 619

show

show

pop

pop

show

push 761

**Đầu ra mẫu 1**

619 928

619 928

TRỐNG

**[DSLK]. Bài 4. queue**

* [**Vấn đề**](https://www.hackerrank.com/contests/28t222kech-lap-898inh-c-contest-12-danh-sach-lien-ket/challenges/dslk-bai-4-hang-doi)
* [**đệ trình**](https://www.hackerrank.com/contests/28t222kech-lap-898inh-c-contest-12-danh-sach-lien-ket/challenges/dslk-bai-4-hang-doi/submissions)
* [**bảng xếp hạng**](https://www.hackerrank.com/contests/28t222kech-lap-898inh-c-contest-12-danh-sach-lien-ket/challenges/dslk-bai-4-hang-doi/leaderboard)
* [**thảo luận**](https://www.hackerrank.com/contests/28t222kech-lap-898inh-c-contest-12-danh-sach-lien-ket/challenges/dslk-bai-4-hang-doi/forum)

Cho hàng đợi lưu 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 hàng đợi nếu hàng đợi không trống, phía trước thì in ra đỉnh ở đầu hàng đợi nếu hàng đợi không trống, queue void, in ra EMPTY

**Định dạng đầu vào**

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

**Hạn chế**

1<=N<=1000;

**Định dạng đầu ra**

In ra câu trả lời của bài toán

**Đầu vào mẫu 0**

9

đẩy 968

pop

đẩy 851

đẩy 51

phía trước

pop

đẩy 159

đẩy 561

đẩy 840

**Đầu ra mẫu 0**

851

**Đầu vào mẫu 1**

7

push 892

pop

pop

pop

pop

phía

trước

**Đầu ra mẫu 1**

Rỗng

Rỗng

**[DSLK]. Bài 5. Sắp xếp DSLK**

* [**Vấn đề**](https://www.hackerrank.com/contests/28t222kech-lap-898inh-c-contest-12-danh-sach-lien-ket/challenges/dslk-bai-5-sap-xep-dslk)
* [**đệ trình**](https://www.hackerrank.com/contests/28t222kech-lap-898inh-c-contest-12-danh-sach-lien-ket/challenges/dslk-bai-5-sap-xep-dslk/submissions)
* [**bảng xếp hạng**](https://www.hackerrank.com/contests/28t222kech-lap-898inh-c-contest-12-danh-sach-lien-ket/challenges/dslk-bai-5-sap-xep-dslk/leaderboard)
* [**thảo luận**](https://www.hackerrank.com/contests/28t222kech-lap-898inh-c-contest-12-danh-sach-lien-ket/challenges/dslk-bai-5-sap-xep-dslk/forum)

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

**Định dạng đầu vào**

Dòng đầu tiên là N : Số lượng nút trong DSLK; Dòng thứ 2 bao gồm N number of the node in DSLK.

**Hạn chế**

1<=N<=100; Các nút trong DSLK là số nguyên âm không vượt quá 1000;

**Định dạng đầu ra**

Dòng 1 trong DSLK được sắp xếp tăng dần; Dòng 2 trong DSLK được sắp xếp giảm dần;

**Đầu vào mẫu 0**

9

59 92 13 13 35 52 17 3 26

**Đầu ra mẫu 0**

3 13 13 17 26 35 52 59 92

92 59 52 35 26 17 13 13 3

**[DSLK]. Bài 6. Đếm tần số**

* [**Vấn đề**](https://www.hackerrank.com/contests/28t222kech-lap-898inh-c-contest-12-danh-sach-lien-ket/challenges/dslk-bai-6-dem-tan-suat)
* [**đệ trình**](https://www.hackerrank.com/contests/28t222kech-lap-898inh-c-contest-12-danh-sach-lien-ket/challenges/dslk-bai-6-dem-tan-suat/submissions)
* [**bảng xếp hạng**](https://www.hackerrank.com/contests/28t222kech-lap-898inh-c-contest-12-danh-sach-lien-ket/challenges/dslk-bai-6-dem-tan-suat/leaderboard)
* [**thảo luận**](https://www.hackerrank.com/contests/28t222kech-lap-898inh-c-contest-12-danh-sach-lien-ket/challenges/dslk-bai-6-dem-tan-suat/forum)

Cho phép 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.

**Định dạng đầu vào**

Bao gồm nhiều dòng, mỗi dòng có thể có nhiều từ.

**Hạn chế**

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

**Định dạng đầu ra**

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

**Đầu vào mẫu 0**

java

python lập trình phụ trợ python 28tech

python

**Đầu ra mẫu 0**

java 1

python 3

phụ trợ 1

lập trình 1

28tech 1

**[DSLK]. Bài 7. Lật ngược DSLK đôi**

* [**Vấn đề**](https://www.hackerrank.com/contests/28t222kech-lap-898inh-c-contest-12-danh-sach-lien-ket/challenges/dslk-bai-7-lat-nguoc-dslk-doi)
* [**đệ trình**](https://www.hackerrank.com/contests/28t222kech-lap-898inh-c-contest-12-danh-sach-lien-ket/challenges/dslk-bai-7-lat-nguoc-dslk-doi/submissions)
* [**bảng xếp hạng**](https://www.hackerrank.com/contests/28t222kech-lap-898inh-c-contest-12-danh-sach-lien-ket/challenges/dslk-bai-7-lat-nguoc-dslk-doi/leaderboard)
* [**thảo luận**](https://www.hackerrank.com/contests/28t222kech-lap-898inh-c-contest-12-danh-sach-lien-ket/challenges/dslk-bai-7-lat-nguoc-dslk-doi/forum)

Cho DSLK đúp chứa các số nguyên, nhiệm vụ của bạn là đảo ngược DSLK kép này và ra các nút trong DSLK sau khi đảo ngược.

**Định dạng đầu vào**

First line is N : number of node in DSLK; Dòng thứ 2 bao gồm N number is numbers in DSLK;

**Hạn chế**

1<=N<=1000; Các nút trong DSLK là số nguyên âm không quá 1000.

**Định dạng đầu ra**

In DSLK double after to reverse.

**Đầu vào mẫu 0**

9

97 207 886 278 987 602 826 817 459

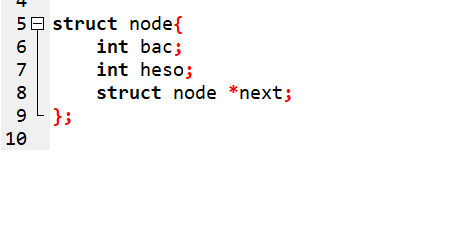
**Đầu ra mẫu 0**

459 817 826 602 987 278 886 207 97

**[DSLK]. Bài 8. DSLK Đa thức**

* [**Vấn đề**](https://www.hackerrank.com/contests/28t222kech-lap-898inh-c-contest-12-danh-sach-lien-ket/challenges/dslk-bai-8-dslk-da-thuc)
* [**đệ trình**](https://www.hackerrank.com/contests/28t222kech-lap-898inh-c-contest-12-danh-sach-lien-ket/challenges/dslk-bai-8-dslk-da-thuc/submissions)
* [**bảng xếp hạng**](https://www.hackerrank.com/contests/28t222kech-lap-898inh-c-contest-12-danh-sach-lien-ket/challenges/dslk-bai-8-dslk-da-thuc/leaderboard)
* [**thảo luận**](https://www.hackerrank.com/contests/28t222kech-lap-898inh-c-contest-12-danh-sach-lien-ket/challenges/dslk-bai-8-dslk-da-thuc/forum)

Cho 2 đa thức A(x) và B(x), tiến trình tính tổng 2 đa thức này và in ra theo bậc thang giảm dần. Use a node in DSLK as after :

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

#include <bits/stdc++.h>

sử dụng không gian tên std ;

nút cấu trúc {

int heso , somu ;

nút \* tiếp theo ;

};

node \* makeNode ( int heso , int somu ){

// cap phat dong

node \* ptr = new node ;

//gan du lieu

ptr -> heso = heso ;

ptr -> somu = somu ;

ptr -> tiếp theo = NULL ;

trả lại 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 ;

}

vô hiệu hóa chúng ( nút \*& đầu , int hs , int sm ){

nút \* temp = đầu ;

nút \* prev = đầu ;

trong khi ( temp != NULL ) {

if ( temp -> somu == sm ) {

temp -> heso += hs ;

trả lại ;

}

trước = tạm thời ;

nhiệt độ = nhiệt độ-> tiếp theo ;

}

nút \* newNode = makeNode ( hs , sm );

nếu ( prev == NULL ){

head = newNode ; trả lại ;

}

trước -> tiếp theo = newNode ;

}

void sx ( nút \*& đầu ) {

for ( nút \* i = đầu ; i != NULL ; i = i -> next ) {

nút \* min = i ;

for ( nút \* j = i -> next ; j != NULL ; j = j -> next ){

if ( min -> somu < j -> somu ){

min = j ;

}

}

hoán đổi ( min -> somu , i -> somu );

hoán đổi ( min -> heso , i -> heso );

}

}

int main (){

nút \* head = NULL ;

chuỗi s ;

trong khi ( cin >> s ){

if ( s != "+" ){

int hs = 0 , sm = 0 ;

int tôi = 0 ;

trong khi ( s [ i ] != 'x' ){

hs = hs \* 10 + ( s [ i] - '0' );

++ tôi ;

}

tôi += 2 ;

trong khi ( i < s . size ()){

sm = sm \* 10 + ( s [ i ] - '0' );

++ tôi ;

}

họ ( đầu , hs , sm );

}

}

sx ( cái đầu );

duyết ( đầu );

}

#include <bits/stdc++.h>

sử dụng không gian tên std ;

nút cấu trúc {

int heso , somu ;

nút \* tiếp theo ;

};

node \* makeNode ( int heso , int somu ){

// cap phat dong

node \* ptr = new node ;

//gan du lieu

ptr -> heso = heso ;

ptr -> somu = somu ;

ptr -> tiếp theo = NULL ;

trả lại 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 ;

}

vô hiệu hóa chúng ( nút \*& đầu , int hs , int sm ){

nút \* temp = đầu ;

nút \* prev = đầu ;

trong khi ( temp != NULL ) {

if ( temp -> somu == sm ) {

temp -> heso += hs ;

trả lại ;

}

trước = tạm thời ;

nhiệt độ = nhiệt độ-> tiếp theo ;

}

nút \* newNode = makeNode ( hs , sm );

nếu ( prev == NULL ){

head = newNode ; trả lại ;

}

trước -> tiếp theo = newNode ;

}

void sx ( nút \*& đầu ) {

for ( nút \* i = đầu ; i != NULL ; i = i -> next ) {

nút \* min = i ;

for ( nút \* j = i -> next ; j != NULL ; j = j -> next ){

if ( min -> somu < j -> somu ){

min = j ;

}

}

hoán đổi ( min -> somu , i -> somu );

hoán đổi ( min -> heso , i -> heso );

}

}

int main (){

nút \* head = NULL ;

chuỗi s ;

trong khi ( cin >> s ){

if ( s != "+" ){

int hs = 0 , sm = 0 ;

int tôi = 0 ;

trong khi ( s [ i ] != 'x' ){

hs = hs \* 10 + ( s [ i] - '0' );

++ tôi ;

}

tôi += 2 ;

trong khi ( i < s . size ()){

sm = sm \* 10 + ( s [ i ] - '0' );

++ tôi ;

}

họ ( đầu , hs , sm );

}

}

sx ( cái đầu );

duyết ( đầu );

}

**Định dạng đầu vào**

Dòng 1 là đa thức A(x).  
Dòng 2 là đa thức B(x).

**Hạn chế**

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

**Định dạng đầu ra**

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

**Đầu vào mẫu 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

**Đầu ra mẫu 0**

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

**[DSLK]. Bài 9. Listing value**

* [**Vấn đề**](https://www.hackerrank.com/contests/28t222kech-lap-898inh-c-contest-12-danh-sach-lien-ket/challenges/dslk-bai-9-liet-ke-gia-tr)
* [**đệ trình**](https://www.hackerrank.com/contests/28t222kech-lap-898inh-c-contest-12-danh-sach-lien-ket/challenges/dslk-bai-9-liet-ke-gia-tr/submissions)
* [**bảng xếp hạng**](https://www.hackerrank.com/contests/28t222kech-lap-898inh-c-contest-12-danh-sach-lien-ket/challenges/dslk-bai-9-liet-ke-gia-tr/leaderboard)
* [**thảo luận**](https://www.hackerrank.com/contests/28t222kech-lap-898inh-c-contest-12-danh-sach-lien-ket/challenges/dslk-bai-9-liet-ke-gia-tr/forum)

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

**Định dạng đầu vào**

First line is N : number of element in DSLK; Dòng thứ 2 bao gồm N number is các phần tử trong DSLK;

**Hạn chế**

1<=N<=1000; Các phần tử trong DSLK là số nguyên không âm 32 bit.

**Định dạng đầu ra**

Trong kết quả của bài toán

**Đầu vào mẫu 0**

6

4 3 2 2 5 5

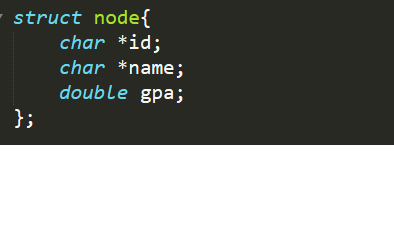
**Đầu ra mẫu 0**

4 3 2 5

**[DSLK]. Bài 10. Điểm trung bình Sinh Viên**

* [**Vấn đề**](https://www.hackerrank.com/contests/28t222kech-lap-898inh-c-contest-12-danh-sach-lien-ket/challenges/dslk-bai-10-gpa-sinh-vien)
* [**đệ trình**](https://www.hackerrank.com/contests/28t222kech-lap-898inh-c-contest-12-danh-sach-lien-ket/challenges/dslk-bai-10-gpa-sinh-vien/submissions)
* [**bảng xếp hạng**](https://www.hackerrank.com/contests/28t222kech-lap-898inh-c-contest-12-danh-sach-lien-ket/challenges/dslk-bai-10-gpa-sinh-vien/leaderboard)
* [**thảo luận**](https://www.hackerrank.com/contests/28t222kech-lap-898inh-c-contest-12-danh-sach-lien-ket/challenges/dslk-bai-10-gpa-sinh-vien/forum)

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

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

#include <bits/stdc++.h>

sử dụng không gian tên std ;

sử dụng ll = dài dài ;

nút cấu trúc {

char \* id ; // id[100]

char \* tên ; // tên[100]

gấp đôi gpa ;

nút cấu trúc \* tiếp theo ; };

nút nút cấu trúc typedef ;

gấp đôi max\_gpa = 0 ;

nút \* makeNode (){

nút \* newNode = nút mới ; newNode -> id = new char [ 100 ]; newNode -> name = new char [ 100 ]; cin . bỏ qua (); được ( newNode -> id ); được ( newNode -> name ); cin >> newNode -> gpa ; max\_gpa =

tối đa ( max\_gpa , newNode -> gpa );

newNode -> next = NULL ;

trả lại nút mới ;

}

void pushBack ( nút \*& đầu ){

nút \* newNode = makeNode ();

nếu ( đầu == NULL ) {

đầu = newNode ;

trả lại ;

}

nút \* temp = đầu ;

trong khi ( temp -> next != NULL ) {

temp = temp -> next ;

}

tạm thời -> tiếp theo = newNode ;

}

int main (){

nút \* head = NULL ;

số nguyên ;

cin >> n ;

trong khi ( n -- ){

pushBack ( đầu );

}

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 ;

}

đầu = đầu -> tiếp theo ;

}

}

**Định dạng đầu vào**

Dòng đầu tiên là N : số lượng sinh viên; Các dòng tiếp theo 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.

**Hạn chế**

1<=N<=1000; Name and id is not too 100 character, GPA is a number ofthực nằm trong khoảng từ 0 đến 4.

**Định dạng đầu ra**

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

**Đầu vào mẫu 0**

7

CNTT1

Vương Đức Phước

2,80

CNTT2

Trần Xuân Phước

1,90

CNTT3

Phạm Xuân Phước

2,90

CNTT4

Đỗ Văn Tuấn

3,80

CNTT5 Vương Hoàng

Hải

2,30

CNTT6

Trần Đức Phước

1,70

CNTT7

Phạm Ngọc Phước

2,40

**Đầu ra mẫu 0**

CNTT4 Đỗ Văn Tuấn 3,80