Status	Finished
Started Saturday, 23 November 2024, 3:46 PM	
Completed	Sunday, 1 December 2024, 10:10 PM
Duration	8 days 6 hours
Marks	120.00/120.00
Grade	10.00 out of 10.00 (100 %)

Question 1 Correct

Mark 10.00 out of 10.00

[InsertXInToPostionY]

Cho dãy số A có n phần tử được đánh số thứ tự từ 1 đến n.

Viết chương trình chèn một số có giá trị là x vào vị trí thứ y của dãy A.

Đầu vào

Đầu vào từ bàn phím gồm ba dòng:

- Dòng đầu tiên chứa số nguyên $n \ (n \le 1000)$;
- ullet Dòng thứ hai chứa n số nguyên là n phần tử của dãy số A, phân tách nhau bởi dấu cách;
- Dòng thứ ba chứa hai số nguyên x và y, phân tách nhau bởi dấu cách.

Đầu ra

In ra màn hình dãy số A sau khi đã chèn số x vào đúng vị trị yêu cầu.

Phân tách các phân tử của dãy số bằng duy nhất một dấu cách.

For example:

Input	Result
5	1 3 5 4 2 5
1 3 4 2 5	
5 3	

```
#include <bits/stdc++.h>
    #define el "\n"
 2
 3
    using namespace std;
 4 v int main() {
 5
        ios_base::sync_with_stdio(0);
 6
        cin.tie(0);
 7
        cout.tie(0);
 8
        int n; cin >> n;
 9
        vector<int> vt;
10
        while(n--)
11 1
            int x; cin >> x;
12
            vt.push_back(x);
13
14
15
        int x, y; cin >> x >> y;
        vt.insert(vt.begin()+y-1,x);
16
17
        for(auto &x : vt) cout << x << " ";</pre>
18
        return 0;
19
    }
20
```

	Input	Expected	Got	
~	5	1 3 5 4 2 5	1 3 5 4 2 5	~
	1 3 4 2 5			
	5 3			
~	10	1 2 3 4 5 6 4 7 8 9 10	1 2 3 4 5 6 4 7 8 9 10	~
	1 2 3 4 5 6 7 8 9 10			
	4 7			
~	20	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	~
	1 2 3 4 5 6 7 8 9 10 11 12 13 14	16 17 18 19 10 20	16 17 18 19 10 20	
	15 16 17 18 19 20			
	10 20			

Correct

Marks for this submission: 10.00/10.00.

1.

Question 2 Correct

Mark 10.00 out of 10.00

[DeletePositionX]

Cho dãy số A có n phần tử được đánh số thứ tự từ 1 đến n.

Viết chương trình xóa phần tử có số thứ tự là x của dãy A.

Đầu vào

Đầu vào từ bàn phím gồm ba dòng:

- Dòng đầu tiên chứa số nguyên $n\ (n \le 1000)$;
- ullet Dòng thứ hai chứa n số nguyên là n phần tử của dãy số A, phân tách nhau bởi dấu cách;
- Dòng thứ ba chứa số nguyên x.

Đầu ra

In ra màn hình các phần tử của dãy A sau khi xóa phần tử ở vị trí x.

Phân tách các phần tử của dãy bởi **một dấu cách duy nhất**.

For example:

Input	Result
5	1 3 2 5
1 3 4 2 5	
3	

```
#include <bits/stdc++.h>
    #define el "\n"
 2
    using namespace std;
 3
 4 v int main() {
 5
        ios_base::sync_with_stdio(0);
 6
        cin.tie(0);
 7
        cout.tie(0);
 8
        int n; cin >> n;
 9
        vector<int> vt;
10
        while(n--)
11
        {
12
            int x; cin >> x;
            vt.push_back(x);
13
14
15
        int y; cin >> y;
16
        vt.erase(vt.begin()+y-1);
        for(auto &x : vt) cout << x << " ";
17
18
        return 0;
19
20
```

	Input	Expected	Got	
~	5	1 3 2 5	1 3 2 5	~
	1 3 4 2 5			
	3			
~	10	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	~
	1 2 3 4 5 6 7 8 9 10			
	10			
~	20	1 2 3 4 5 6 7 8 9 11 12 13 14 15 16	1 2 3 4 5 6 7 8 9 11 12 13 14 15	~
	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	17 18 19 20	16 17 18 19 20	
	16 17 18 19 20			
	10			

Correct

Marks for this submission: 10.00/10.00.

1.

```
Question 3
Correct
Mark 10.00 out of 10.00
```

[AscendingSorted]

Viết chương trình sắp xếp mảng gồm n số nguyên theo chiều tăng dần.

Đầu vào

Đầu vào từ bàn phím gồm hai dòng:

- Dòng đầu tiên chứa số nguyên $n \ (n < 100)$;
- ullet Dòng thứ hai chứa n số nguyên là các phần tử của mảng, phân tách nhau bởi dấu cách.

Đầu ra

In ra màn hình các phần tử của mảng sau khi sắp xếp, phân tách chúng bởi duy nhất một dấu cách.

For example:

Input	Result
5	2 4 5 6 7
5 7 2 6 4	

```
#include <bits/stdc++.h>
    #define el "\n"
 3 using namespace std;
4 ▼
    int main() {
 5
        ios_base::sync_with_stdio(0);
        cin.tie(0);
 6
 7
        cout.tie(0);
 8
        int n; cin >> n;
        vector<int> vt;
 9
10
        while(n--)
11 ,
12
            int x; cin >> x;
13
            vt.push_back(x);
14
        sort(vt.begin(),vt.end());
15
        for(auto &x : vt) cout << x << " ";</pre>
16
17
        return 0;
18
19
```

	Input	Expected	Got	
~	5 7 2 6 4	2 4 5 6 7	2 4 5 6 7	~
~	10 10 9 8 7 6 5 4 3 2 1	1 2 3 4 5 6 7 8 9 10	1 2 3 4 5 6 7 8 9 10	~

Correct

Question 4 Correct Mark 10.00 out of 10.00

[DescendingSorted]

Viết chương trình sắp xếp mảng gồm n số nguyên theo chiều giảm dần.

Đầu vào

Đầu vào từ bàn phím gồm hai dòng:

- Dòng đầu tiên chứa số nguyên $n \ (n < 100)$;
- ullet Dòng thứ hai chứa n số nguyên là các phần tử của mảng, phân tách nhau bởi dấu cách.

Đầu ra

In ra màn hình các phần tử của mảng sau khi sắp xếp, phân tách chúng bởi duy nhất một dấu cách.

For example:

Input	Result
5	7 6 5 4 2
5 7 2 6 4	

```
#include <bits/stdc++.h>
   #define el "\n"
 3 using namespace std;
4 ▼
    int main() {
 5
        ios_base::sync_with_stdio(0);
 6
        cin.tie(0);
 7
        cout.tie(0);
 8
        int n; cin >> n;
        vector<int> vt;
 9
10
        while(n--)
11 ,
12
            int x; cin >> x;
13
            vt.push_back(x);
14
        sort(vt.begin(),vt.end(),greater<int>());
15
        for(auto &x : vt) cout << x << " ";</pre>
16
17
        return 0;
18
19
```

	Input	Expected	Got	
~	5	7 6 5 4 2	7 6 5 4 2	~
	5 7 2 6 4			
~	10	10 9 8 7 6 5 4 3 2 1	10 9 8 7 6 5 4 3 2 1	~
	1 2 3 4 5 6 7 8 9 10			

Correct

Question 5 Correct Mark 10.00 out of 10.00

[MixedSort1]

Cho một mảng gồm n số nguyên.

Viết chương trình sắp xếp lại các phần tử trong mảng này, sao cho sau khi sắp xếp:

- tất cả các phần tử chẵn nằm ở phía trái của mảng;
- tất cả các phần tử lẻ nằm ở phía phải của mảng;
- các phần tử chẵn sắp xếp theo chiều tăng dần;
- các phần tử lẻ sắp xếp theo chiều giảm dần.

Đầu vào

Đầu vào từ bàn phím gồm hai dòng:

- Dòng đầu tiên chứa số nguyên $n \ (n \le 100)$;
- ullet Dòng tiếp theo chứa n số nguyên là các phần tử của mảng. Các số được phân tách nhau bởi một dấu cách.

Đầu ra

In ra màn hình các phần tử của mảng sau khi sắp xếp, phân tách các phần tử bởi duy nhất một dấu cách.

For example:

Input	Result
7	2 4 6 8 9 7 5
5 9 2 8 6 4 7	

```
#include <bits/stdc++.h>
    #define el "\n"
 3
    using namespace std;
 4
    int main() {
        ios_base::sync_with_stdio(0);
 5
 6
        cin.tie(0);
 7
        cout.tie(0);
 8
        int n; cin >> n;
 9
        vector<int> vtc,vtl;
10
        while(n--)
11
        {
12
             int x; cin >> x;
13
             x % 2 ? vtl.push_back(x) : vtc.push_back(x);
14
15
        sort(vtc.begin(),vtc.end());
        sort(vtl.begin(),vtl.end(),greater<int>());
16
17
        for(auto &x : vtc) cout << x <<" ";</pre>
        for(auto &x : vtl) cout << x <<" ";</pre>
18
19
        return 0;
20
    }
21
```

	Input	Expected	Got	
~	7	2 4 6 8 9 7 5	2 4 6 8 9 7 5	~
	5 9 2 8 6 4 7			
~	10	2 4 6 8 10 9 7 5 3 1	2 4 6 8 10 9 7 5 3 1	~
	1 2 3 4 5 6 7 8 9 10			
~	10	0 2 4 6 8 48 33 9 3 1	0 2 4 6 8 48 33 9 3 1	~
	1 3 0 8 4 6 2 33 9 48			

Correct

Marks for this submission: 10.00/10.00.

1.

Question 6 Correct Mark 10.00 out of 10.00

[FastestFinger]

Ai là triệu phú là phiên bản Tiếng Việt của Trò chơi truyền hình nổi tiếng của Anh *Who Wants to Be a Millionaire?*, do VTV và Mesa thực hiện. Chương trình bắt đầu được phát sóng từ ngày 4 tháng 1 năm 2005 và Người dẫn chương trình là nhà báo Lại Văn Sâm. Mục đích chính của trò chơi là giành chiến thắng bằng cách trả lời chính xác 15 câu hỏi của chương trình.

Những người chơi có mặt trong chương trình phát sóng, trước tiên phải tham gia một vòng chơi loại nhỏ gọi là **Fastest Finger First** (tạm dịch là **Gõ Bàn Phím Nhanh Nhất**) hay **Fastest Finger** (tạm dịch là **Gõ Bàn Phím Nhanh** hay **Bấm Bàn Phím Nhanh**). Người chơi sẽ nhận được một câu hỏi với bốn phương án từ người dẫn chương trình. Nhiệm vụ của người chơi là sắp xếp các phương án theo một trật tự nhất định được quy định trong câu hỏi. Người chơi sắp xếp đúng các phương án trong thời gian nhanh nhất trong khoảng thời gian quy định sẽ trở thành Người Chơi Chính ngồi trên "Ghế Nóng" đối diện ghế ngồi của dẫn chương trình ở giữa sân khấu.

Vậy bạn đã sẵn sàng để trở thành triệu phú? Hãy bắt đầu bằng việc sắp xếp các cụm từ dưới đây theo thứ tự xuất hiện trong từ điển.

Đầu vào: 4 cụm từ xuất hiện trên bốn dòng tương ứng với các phương án: A, B, C và D.

Đầu ra: thứ tự đúng của các phương án, cách nhau bởi dấu cách.

For example:

Input	Result
May	C D A B
Troi	
Cua	
Em	

```
#include <bits/stdc++.h>
    #define el "\n"
 2
    #define fi fisrt
 3
    #define se second
    using namespace std;
 6 v int main() {
 7
        ios_base::sync_with_stdio(0);
 8
        cin.tie(∅);
 9
        cout.tie(0);
10
        string s1, s2, s3, s4;
11
        //cin >> s1 >> s2 >> s3 >> s4;
12
        getline(cin, s1);
13
        getline(cin, s2);
        getline(cin, s3);
14
15
        getline(cin, s4);
16
        vector<pair<string,char>>vt;
17
        vt.push_back({s1,'A'});
18
        vt.push_back({s2,'B'});
19
        vt.push_back({s3,'C'});
20
        vt.push_back({s4,'D'});
21
        sort(vt.begin(),vt.end());
22
        for(auto &x : vt) cout << x.se << " ";</pre>
23
        return 0;
24
25
```

May Troi Cua Em	CDAB	CDAB	~
Cua			
Em			
LIII			
Hoa hong	ABDC	ABDC	~
hoa hue			
hoa lan			
hoa huong duong			
Rau den	DACB	DACB	~
rau day			
Rau muong			
ca rot			
Thu hai	CABD	CABD	~
Thu ba			
Chu nhat			
Thu bay			
1	CADB	CADB	~
3			
0			
2			
	hoa hue hoa lan hoa huong duong Rau den rau day Rau muong ca rot Thu hai Thu ba Chu nhat Thu bay 1 3	hoa hue hoa lan hoa huong duong Rau den rau day Rau muong ca rot Thu hai Chu nhat Thu bay 1 C A D B 3	hoa hue hoa lan hoa huong duong Rau den rau day Rau muong ca rot Thu hai Chu nhat Thu bay C A D B C A D B C A D B C A D B C A D B C A D B

Correct

Marks for this submission: 10.00/10.00.

1.

Question **7**

Correct

Mark 10.00 out of 10.00

[FindingNemo]

Nemo được mời tới tham dự hội nghị bàn tròn giữa các nhân vật hoạt hình. Tất cả các khách mời sẽ được gửi thiệp mời kèm theo một danh sách khách mời.

Nemo muốn biết các nhân vật ngồi cạnh mình là ai để có thể chuẩn bị quà cho họ. Hãy giúp Nemo tìm tên của các nhân vật này, biết rằng tên trong danh sách khách mời được sắp xếp theo vị trí ngồi.

Đầu vào: số nguyên \(n \) số lượng khách mời trong danh sách trên một dòng, \(n \) dòng tiếp theo là tên của các vị khách mời. Đầu ra: Tên của hai người ngồi cạnh Nemo theo định dạng "<name1> and <name2>" (name1 là tên của người đứng trước và name2 là tên của người đứng sau Nemo trong danh sách).

For example:

Input	Result
5	Aurora and Anastatia
Anastatia	
Alice	
Ariel	
Aurora	
Nemo	

```
#include <bits/stdc++.h>
    #define el "\n"
 2
 3
    #define fi fisrt
 4
    #define se second
 5
    using namespace std;
 6 ,
    int main() {
        ios_base::sync_with_stdio(0);
 7
 8
        cin.tie(0);
 9
        cout.tie(0);
10
        int n;
11
        cin >> n;
12
        cin.ignore();
        int id = 0;
13
14
        vector<string>vt;
        for(int i=1;i<=n;i++)</pre>
15
16
        {
17
             string s;
             getline(cin,s);
18
             if(s=="Nemo")
19
20
             {
21
                 id=i;
22
23
             vt.push_back(s);
24
25
         vt.insert(vt.begin(),vt[n-1]);
        vt.push_back(vt[1]);
26
         cout<<vt[id-1]<<" and " <<vt[id+1];</pre>
27
28
         return 0;
29
30
```

	Input	Expected	Got	
~	5	Aurora and Anastatia	Aurora and Anastatia	~
	Anastatia			
	Alice			
	Ariel			
	Aurora			
	Nemo			
~	10	Bert and Aurora	Bert and Aurora	~
	Big Al			
	Belle			
	Bert			
	Nemo			
	Aurora			
	Alice			
	Beast			
	Anastatia			
	Ariel			
	Baloo			
~	20	Baloo and Beast	Baloo and Beast	~
	Bert			
	Alice			
	Cinderella			
	Big Al			
	Baloo			
	Nemo			
	Beast			
	Brer Rabbit			
	Brer Bear			
	Clopin			
	Cinderella's Mice			
	Aurora			
	Brer Fox			
	Belle			
	Bullseye			
	Chewbaka			
	Anastatia			
	Ariel			
	Captain Hook			
	Buzz			

	Input	Expected	Got	
<u> </u>	30	Captain Hook and Esmeralda	Captain Hook and Esmeralda	~
	Brer Rabbit			
	Donald Duck			
	Cruella De Ville			
	Cinderella			
	Captain Hook			
	Nemo			
	Esmeralda			
	Beast			
	Alice			
	Cinderella's Mice			
	Aurora			
	Chewbaka			
	Evil Queen			
	Belle			
	Bullseye			
	Anastatia			
	Buzz			
	Clopin			
	Daisy Duck			
	Bert			
	Eeyore			
	Brer Bear			
	Ariel			
	Dwarfs			
	Fairy Godmother			
	Brer Fox			
	Baloo			
	Ewoks			
	Drizella			
	Big Al			

	Input	Expected	Got	
/	50	Green Army Men and Goofy	Green Army Men and Goofy	~
	Aurora			
	Eeyore			
	Cinderella			
	Evil Queen			
	Friar Tuck			
	Green Army Men			
	Nemo			
	Goofy			
	Esmeralda			
	Francis Ladybug			
	Ariel			
	Frollo			
	Bert			
	Geppetto			
	Buzz			
	Brer Bear			
	Flik			
	Bullseye			
	Clopin			
	Chewbaka			
	Jessie			
	Kida			
	Alice			
	Beast			
	Anastatia			
	Jane Porter			
	J.Worthington Foulfellow			
	Jasmine			
	Hercules			
	Brer Rabbit			
	Captain Hook			
	Jafar			
	Cinderella's Mice			
	Big Al			
	Gov Radcliffe			
	Cruella De Ville			
	Drizella			
	King Louie			
	Brer Fox			
	Donald Duck			
	Ewoks			
	Jiminy Cricket			
	Gold Dust Pete			
	Baloo			
	Genie			
	Dwarfs			
	Gideon			
	Belle			

Input	Expected	Got	
Daisy Duck			
Fairy Godmother			

Correct

```
Question 8
Correct
Mark 10.00 out of 10.00
```

[FindOccurrences]

Các trình duyệt web thường có một chức năng tìm kiếm một từ trong một trang web. Chẳng hạn khi bạn tìm kiếm từ "UET" trên trang UETCodeHub, trình duyệt sẽ hiện ra có một từ "UET". Hãy viết một chương trình để xác định xem có bao nhiêu từ xuất hiện trong một câu.

Đầu vào: Một chuỗi kí tự trên một dòng. Từ khóa cần tìm ở dòng tiếp theo.

Đầu ra: Số lần xuất hiện của từ tìm kiếm.

For example:

Input	Result
There's a fire that burns inside.	1
fire	

```
#include <iostream>
 1
 2
    #include <string>
 4
    using namespace std;
 5
 6 v int main() {
 7
        string sentence, keyword;
 8
        getline(cin, sentence);
 9
        getline(cin, keyword);
10
        int count = 0;
11
        size_t pos = sentence.find(keyword);
12 •
        while (pos < sentence.size()) {</pre>
13
             pos = sentence.find(keyword, pos + keyword.length());
14
15
16
17
        cout << count << endl;</pre>
18
19
        return 0;
20
21
```

	Input	Expected	Got	
~	There's a fire that burns inside.	1	1	~
	fire			

	Input	Expected	Got	
~	It's an instinct that never lies. The target's tattooed between our eyes. never lies	1	1	~
~	Everyone's watching through your eyes. There's only two options. Win or die, win or die. Win or die, die, die, die, die. die	7	7	~
~	It went from a spark to an open flame. Now destiny's calling out your name. So reply, so reply.	0	0	~
~	And ignite. Ignite. Ignite. Ignite	3	3	~

Correct

```
Question 9

Correct

Mark 10.00 out of 10.00
```

[InsertWithoutChangingOrder]

Cho một mảng gồm \(n \) số thực đã được sắp xếp theo thứ tự tăng dần.

Viết chương trình chèn một số thực \(k \) vào mảng trên sao cho thứ tự sắp xếp không đổi. In dãy mới ra màn hình.

Kết quả làm tròn đến số thập phân thứ hai.

For example:

```
#include <bits/stdc++.h>
 2
    using namespace std;
    #define 11 long long
 3
    #define el "\n"
 4
 5
6 int main()
 7 ▼ {
 8
        ios_base::sync_with_stdio(0);
 9
        cin.tie(0); cout.tie(0);
10
        int n; cin >> n;
        double a[n+1];
11
12
        for(int i = 1; i <= n; i++) cin >> a[i];
        double x; cin >> x;
13
14
        int ok = 0;
        for(int i =1; i <= n; i++)</pre>
15
16
17
            if(x < a[i] && ok ==0) cout << fixed << setprecision(2) << x << " ", ok =1;
18
            cout << fixed << setprecision(2) << a[i] << " ";</pre>
19
        }
20
21
        return 0;
22
23
```

	Input	Expected	Got	
/	5	-2.51 -1.62 0.97 2.13 7.57	-2.51 -1.62 0.97 2.13	~
	-1.61996539202 0.968616409208 2.12587576231 7.56980530885	11.31	7.57 11.31	
	11.305029863			
	-2.50635869654			
<u> </u>	5	-8.08 -6.47 -5.14 4.93	-8.08 -6.47 -5.14 4.93	~
	-8.08076929787 -6.47268489558 -5.14371884832	5.37 6.78	5.37 6.78	
	5.37159276481 6.77924146934			
	4.92793607475			
	100	-29.32 -20.92 -19.57	-29.32 -20.92 -19.57	ļ
	-29.3235478438 -20.9220576434 -19.5707065441	-17.66 -16.60 -15.57	-17.66 -16.60 -15.57	
	-17.6587017695 -16.6010112667 -15.5650729808	-15.13 -14.69 -14.57	-15.13 -14.69 -14.57	
	-15.1277832776 -14.688239386 -14.5710199344	-14.36 -13.98 -13.77	-14.36 -13.98 -13.77	
	-14.3617535799 -13.9842876915 -13.7722289965	-13.60 -12.98 -12.60	-13.60 -12.98 -12.60	
	-13.6004178886 -12.9784142201 -12.5985063419	-11.93 -11.78 -10.94	-11.93 -11.78 -10.94	
	-11.9309728003 -11.7830388343 -10.9392858707	-10.57 -10.33 -9.75 -9.71	-10.57 -10.33 -9.75	
	-10.5730993344 -10.3319478242 -9.7529324221	-9.32 -9.30 -8.90 -8.46	-9.71 -9.32 -9.30 -8.90	
	-9.70585956233 -9.31904905615 -9.29853023349	-8.05 -8.03 -7.95 -7.88	-8.46 -8.05 -8.03 -7.95	
	-8.90478964495 -8.45819998488 -8.04599004429	-7.82 -7.60 -7.48 -7.35	-7.88 -7.82 -7.60 -7.48	
	-8.0346153743 -7.94926280505 -7.88434110905 -7.823574344	-7.30 -5.55 -5.28 -4.98	-7.35 -7.30 -5.55 -5.28	
	-7.60233278572 -7.47951644975 -7.34978811616	-4.79 -4.52 -4.08 -3.06	-4.98 -4.79 -4.52 -4.08	
	-7.29530537726 -5.5483291453 -5.27645382851	-3.04 -3.04 -3.00 -2.73	-3.06 -3.04 -3.04 -3.00	
	-4.97720790435 -4.79224390662 -4.51593634541	-2.40 -2.24 -2.07 -2.05	-2.73 -2.40 -2.24 -2.07	
	-4.08172642756 -3.05511040257 -3.04160723482	-1.96 -1.84 -0.95 -0.93	-2.05 -1.96 -1.84 -0.95	
	-3.03611935862 -3.00474136759 -2.73359106019	-0.70 -0.08 0.29 0.30 0.65	-0.93 -0.70 -0.08 0.29	
	-2.4033742979 -2.24439627251 -2.06949064853	0.68 0.99 1.11 1.19 1.34	0.30 0.65 0.68 0.99 1.11	
	-2.05070763335 -1.962416675 -1.84007969027	1.56 1.62 1.63 1.80 2.81	1.19 1.34 1.56 1.62 1.63	
	-0.950603836211 -0.93295132839 -0.703545865189	2.90 2.97 3.96 4.03 4.09	1.80 2.81 2.90 2.97 3.96	
	-0.0813846512361 0.294717489155 0.300739300058	4.26 4.55 4.90 5.11 5.43	4.03 4.09 4.26 4.55 4.90	
	0.651433519179 0.680020167494 0.985637478402	5.92 6.29 6.80 7.01 8.94	5.11 5.43 5.92 6.29 6.80	
	1.10872370933 1.18927209797 1.33507232145 1.56466328236	9.26 9.35 9.84 10.09 10.19	7.01 8.94 9.26 9.35 9.84	
	1.61662772067 1.62655254121 1.79536821921 2.81047748431	10.43 10.95 12.29 12.38	10.09 10.19 10.43 10.95	
	2.90318009804 2.97287247349 3.95592581071 4.03032324789	13.39 13.48 13.59 15.04	12.29 12.38 13.39 13.48	
	4.08946194936 4.25715883418 4.54529057333 4.90180655733	17.82 19.06 20.87 36.50	13.59 15.04 17.82 19.06	
	5.10558105843 5.42592332157 5.91833635651 6.28899263404		20.87 36.50	
	6.79682572042 7.00646995056 8.94220748137 9.26186175761			
	9.35445909702 9.8367929271 10.0933909647 10.1878738423			
	10.4279078876 10.9471713072 12.2891551075 13.3870984366			
	13.4821701755 13.5872477108 15.0413983707 17.8201710401			
	19.059716493 20.8703813875 36.4983253209			
	12.384094591			

	Input	Expected	Got	
/	49	-23.54 -12.06 -11.59	-23.54 -12.06 -11.59	~
	-23.5431900808 -12.0588613104 -11.5916064405	-11.57 -11.09 -10.07 -9.69	-11.57 -11.09 -10.07	
	-11.5701991792 -11.0906962311 -9.68567663184	-8.95 -8.87 -8.48 -8.45	-9.69 -8.95 -8.87 -8.48	
	-8.95459247853 -8.87377773467 -8.47757954995	-7.92 -7.53 -6.96 -6.55	-8.45 -7.92 -7.53 -6.96	
	-8.44870030534 -7.91582050361 -7.52596210494	-6.53 -6.14 -5.46 -5.20	-6.55 -6.53 -6.14 -5.46	
	-6.95999263338 -6.54794815863 -6.53130176386	-5.05 -3.73 -3.60 -2.64	-5.20 -5.05 -3.73 -3.60	
	-6.14362476219 -5.46374607615 -5.19706371846	-2.64 -2.60 -1.71 -0.81	-2.64 -2.64 -2.60 -1.71	
	-5.05200211281 -3.73251974818 -3.59699418411	-0.23 -0.20 0.52 1.16 2.09	-0.81 -0.23 -0.20 0.52	
	-2.6409843316 -2.63934102685 -2.5966587278 -1.71257031964	3.96 4.07 4.67 4.73 4.99	1.16 2.09 3.96 4.07 4.67	
	-0.813287597101 -0.228237328881 -0.200463094387	5.32 6.13 6.45 8.25 8.62	4.73 4.99 5.32 6.13 6.45	
	0.524665045565 1.15903999384 2.09015526815 3.96450956993	9.00 9.13 9.85 10.06 11.32	8.25 8.62 9.00 9.13 9.85	
	4.06726655975 4.67173176886 4.73155142239 4.99203697466	12.07 12.67 23.77	10.06 11.32 12.07 12.67	
	5.32213332981 6.13138630965 6.45036220859 8.24976236845		23.77	
	8.61593274178 8.99573358031 9.13352011411 9.84991759956			
	10.0608796271 11.3194883709 12.0711570309 12.6698684794			
	23.7731949094			
	-10.0723885197			
,	65	-324.73 -205.12 -185.03	-324.73 -205.12 -185.03	
	-324.729422285 -205.117186089 -185.029923741	-140.31 -137.58 -119.60	-140.31 -137.58 -119.60	
	-140.310672543 -137.576516757 -119.601870379	-114.16 -100.24 -97.44	-114.16 -100.24 -97.44	
	-114.162538139 -97.4425888244 -93.4140328426	-93.41 -71.56 -67.34	-93.41 -71.56 -67.34	
	-71.5632454406 -67.3385346946 -61.9520278881	-61.95 -55.85 -53.62	-61.95 -55.85 -53.62	
	-55.8514149689 -53.6208457955 -52.6385388995	-52.64 -50.27 -22.98	-52.64 -50.27 -22.98	
	-50.2654503194 -22.9849096872 -13.1087506386	-13.11 -12.98 -10.75 -7.28	-13.11 -12.98 -10.75	
	-12.9808928182 -10.7498135708 -7.27846699963	-6.57 -5.65 -3.92 1.69	-7.28 -6.57 -5.65 -3.92	
	-6.56697532694 -5.65005854091 -3.92387155238	1.73 11.26 15.14 18.53	1.69 1.73 11.26 15.14	
	1.68774719468 1.72730718857 11.2579918837 15.1387028491	20.05 21.52 22.95 23.45	18.53 20.05 21.52 22.95	
	18.5261412224 20.0538437757 21.5180770085 22.9549659059	27.51 30.23 37.95 39.52	23.45 27.51 30.23 37.95	
	23.4504225558 27.5104345706 30.2306225848 37.9454501842	41.28 41.46 53.43 56.93	39.52 41.28 41.46 53.43	
	39.5241527785 41.2805600914 41.4619914647 53.4344233531	60.30 62.42 63.77 64.53	56.93 60.30 62.42 63.77	
	56.9268111754 60.301960241 62.4219845083 63.7712491599	66.82 71.13 72.31 74.32	64.53 66.82 71.13 72.31	
	64.532227855 66.8217887735 71.1307906739 72.3081025118	74.37 74.63 76.77 81.26	74.32 74.37 74.63 76.77	
	74.3199925801 74.3708878858 74.6250885183 76.7729262904	81.56 87.05 88.02 108.57	81.26 81.56 87.05 88.02	
	81.258650039 81.5581468751 87.0510685529 88.0174466823	114.13 128.58 134.90	108.57 114.13 128.58	
	108.572418609 114.132215067 128.580077286 134.901479331	141.71 148.80 187.45	134.90 141.71 148.80	
	141.708696355 148.798666614 187.454440084 229.622880554	229.62 257.66	187.45 229.62 257.66	
	257.659436981		25,145 225,02 257,00	
	-100.237283266			

Correct

```
Question 10
Correct
Mark 10.00 out of 10.00
```

[LongestHarmoniousSubsequence]

Dãy hài hòa là một dãy thỏa mãn điều kiện giá trị lớn nhất và giá trị nhỏ nhất của dãy khác nhau đúng bằng \(1 \).

Viết hàm int findLHS(int arr[], int n) trả về độ dài của dãy hài hòa con dài nhất có thể có trong mảng giá trị nguyên \(arr \) có \(n \) phần tử

Ví dụ:

Input: 13225237

Output: 5 - Dãy hài hòa con dài nhất là [3,2,2,2,3]

For example:

Test	Input	Result
findLHS(arr, n)	4	2
	1 2 3 4	

Answer: (penalty regime: 0 %)

```
#include <bits/stdc++.h>
 2
    using namespace std;
    #define 11 long long
    #define ull unsigned long long
    #define el "\n"
 5
    const int mod=1e9+7;
 6
 8
    int findLHS(int a[], int n)
 9 🔻
    {
10
         int ans = 1;
11
        map<int,int>mp;
        vector<int>vt;
12
13
        for(int i = 0; i < n; i++)</pre>
14
15
             mp[a[i]]++;
16
             if(mp[a[i]] == 1) vt.push_back(a[i]);
17
18
         sort(vt.begin(), vt.end());
19
        int m = vt.size() - 1;
20
         for(int i = 0; i < m;i++)</pre>
21 ,
             ans = max(ans, mp[vt[i]]);
22
23
             if(abs(vt[i] - vt[i+1]) == 1)
24
             {
25
                 //cout << vt[i] <<" "<< vt[i+1]<<" "<< mp[vt[i]]+mp[vt[i+1]]<<el;
26
27
                 ans = max(ans, mp[vt[i]]+mp[vt[i+1]]);
            }
28
29
30
        return ans;
31
32
    void print()
33
34 ▼
    {
        int n; cin >> n;
35
36
         int a[n];
37
        for(auto &x : a) cin >> x;
38
         cout << findLHS(a, n);</pre>
39
40
```

https://dev.uet.vnu.edu.vn/mod/quiz/review.php?attempt=152751&cmid=4829

	Test	Input	Expected	Got	
~	findLHS(arr,	4	2	2	~
	n)	1 2 3 4			
~	findLHS(arr,	25	2	2	~
	n)	27 20 17 32 7 49 19 52 26 1 16 4 47 42 49 47 35 51 4 8 40 15 45 39 12			
~	findLHS(arr,	50	6	6	~
	n)	15 30 27 17 0 31 37 42 47 18 22 37 48 0 41 23 1 1 5 14 36 41 18 46 29 21 26 35			
		27 46 44 22 22 43 38 2 46 0 15 22 8 24 12 34 21 37 11 39 17 48			
~	findLHS(arr,	150	6	6	~
	n)	6 26 15 84 26 99 17 99 2 14 63 56 97 61 57 0 4 15 82 60 96 47 24 46 36 25 49 54			
		4 55 62 89 70 12 49 36 85 93 29 96 17 70 36 52 83 44 0 7 23 50 44 36 42 97 42 28			
		44 17 21 18 16 66 69 40 42 17 91 3 2 95 23 21 79 23 4 8 63 72 96 58 68 86 79 36			
		79 93 30 51 87 29 11 58 48 57 52 68 20 43 63 65 92 10 16 7 61 64 52 27 50 59 46			
		1 19 65 42 11 85 25 4 14 78 13 38 46 20 86 47 22 39 86 69 35 79 74 88 85 12 93			
		88 71 77 52 94 66 29 27 25 93 99 49			
~	findLHS(arr,	250	15	15	~
	n)	37 2 44 4 6 40 9 3 31 18 8 13 12 16 18 34 16 44 35 42 46 42 26 12 33 5 46 6 35			
		14 51 9 53 24 17 29 29 37 49 10 52 38 50 46 8 27 26 19 29 35 6 36 30 19 16 28 37			
		41 53 27 4 18 35 4 43 33 29 46 41 15 22 50 18 30 0 0 7 42 10 42 36 41 29 6 47 30			
		3 24 44 25 22 52 29 1 26 19 48 33 39 51 33 7 9 47 30 18 27 21 46 2 51 37 19 44			
		51 51 13 8 50 4 26 28 15 28 22 34 32 31 12 10 10 47 2 23 21 46 18 5 54 21 48 40			
		18 4 21 2 43 36 3 54 31 52 24 49 37 30 45 32 15 20 4 32 25 28 6 5 36 43 21 54 19			
		33 17 1 10 25 12 51 33 30 18 44 14 50 1 11 35 43 31 35 15 52 39 36 53 13 36 49 0			
		39 39 31 34 4 44 5 17 29 12 39 8 43 39 49 43 16 10 24 14 54 42 54 0 1 34 50 41			
		29 47 23 39 35 14 41 21 5 41 50 5 4 33 1 2 35 13 46 5 10 1 25			

Correct

Question 11 Correct Mark 10.00 out of 10.00

[Permutation]

Viết chương trình nhận đầu vào là một bộ hoán vị của các số từ \(0 \) đến \(4 \) và in ra màn hình checkerboard tương ứng với hoán vị đó.

Ví dụ với bộ hoán vị đầu vào là { \(4, 1, 3, 0, 2 \) }, ta có checkerboard tương ứng là

```
****Q
*Q***

****
```

0

For example:

Input	Result
4 3 2 1 0	* * * * Q
	* * * Q *
	* * Q * *
	* Q * * *
	Q * * * *

```
#include <bits/stdc++.h>
 2
    using namespace std;
    #define ll long long
 3
    #define ull unsigned long long
 5
    #define el "\n"
 6
    const int mod=1e9+7;
 7
 8
    int main()
 9 •
         ios_base::sync_with_stdio(0);
10
         cin.tie(0); cout.tie(0);
11
12
         int a[5];
13
         for(auto &x : a) cin >> x;
14
         for(int i = 0; i <= 4; i++)
15
             for(int j = 0; j <= 4; j++)</pre>
16
17
18
                 if(a[i] == j) cout << "Q";</pre>
19
                 else cout <<"*";</pre>
20
                 cout << " ";
21
22
             cout << el;</pre>
         }
23
24
         return 0;
25
26
```

	Input	Expected	Got	
~	4 3 2 1 0	* * * * Q	* * * * Q	~
		* * * Q *	* * * Q *	
		* * Q * *	* * Q * *	
		* Q * * *	* Q * * *	
		Q * * * *	Q * * * *	
~	2 1 4 3 0	* * Q * *	* * Q * *	~
		* Q * * *	* Q * * *	
		* * * * Q	* * * * Q	
		* * * Q *	* * * Q *	
		Q * * * *	Q * * * *	
~	4 1 3 0 2	* * * * Q	* * * * Q	~
		* Q * * *	* Q * * *	
		* * * Q *	* * * Q *	
		Q * * * *	Q * * * *	
		* * Q * *	* * Q * *	

Correct

Marks for this submission: 10.00/10.00.

https://dev.uet.vnu.edu.vn/mod/quiz/review.php?attempt=152751&cmid=4829

Question 12 Correct Mark 10.00 out of 10.00

[SevenNumber]

ắt hẳn có đấng siêu nhiên nào đó đã tạo ra con số \(7 \) diệu kỳ. Nó đã xuất hiện khắp nơi trong cuộc sống.

Trong Phật giáo. Khi sinh ra, Đức Phật bước $\ (7\)$ bước, nở ra $\ (7\)$ đóa hoa sen. Lúc chết, con người ta phải xuống $\ (7\)$ tầng địa ngục và để cúng cho họ, người ta lấy bội số của số $\ (7\)$ = $\ (49\)$ ngày.

Trong Thiên Chúa giáo, Chúa Trời đã mất \(7 \) ngày để sáng tạo nên vũ trụ. Thiên Chúa lấy xương sườn số \(7 \) bên trái của Adam để tạo ra Eva vì nó gần cánh tay và trái tim của người đàn ông nên được che chở và yêu thương.

Một tuần lễ có (7) ngày, nghệ thuật có (7) ngành, âm nhạc có (7) nốt, văn minh nhân loại có (7) kỳ quan thế giới mới. Cùng với (7) giai đoạn tiến hóa, con người có bảy cái lỗ trên mặt, gọi là thất khiếu (hai mắt, hai tai, hai lỗ mũi, và miệng) và (7) trạng thái tình cảm khác nhau, gọi là thất tình (ái, ố, hỉ, nộ, lạc, ai, dục). Ngưu Lang, Chức Nữ gặp nhau ngày (7) tháng (7). Những thứ quý báu nhất đối với mọi người là thất bảo (vàng bạc, ngọc, hổ phách, mã não, xà cừ, san hô và lưu ly),...

Hãy viết một chương trình tìm kiếm tất cả các số \(7 \) có trong một dãy số.

Đầu vào: số nguyên \(n \) trên một dòng và \(n \) số nguyên ở dòng kế tiếp.

Đầu ra:

- Vị trí của các số \(7 \) trong dãy cách nhau bởi dấu cách, vị trí lớn hơn đứng trước và bắt đầu tính từ \(0\).
- Nếu không tìm được số \(7 \) nào in ra "Not found"

For example:

Input	Result
5	0
7 10 -2 12 -1	
10	Not found
-12 -5 -4 -12 -14 5 10 -8 4 -3	

```
#include <bits/stdc++.h>
    using namespace std;
    #define ll long long
 3
    #define ull unsigned long long
    #define el "\n"
    const int mod=1e9+7;
 6
 7
 8
    int main()
 9 •
        ios base::sync with stdio(0);
10
11
        cin.tie(0); cout.tie(0);
         int n;
12
13
         cin >> n;
        int a[n];
14
         vector<int> vt;
15
16
         for(int i = 0; i < n; i++)
17
18
             cin >> a[i];
             if(a[i] == 7) vt.push_back(i);
19
20
21
         if(vt.size() == 0)
22
             cout << "Not found";</pre>
23
24
             return 0;
25
         sort(vt.begin(),vt.end(), greater<int>());
26
```

```
27 | for(auto &x : vt) cout << x << " ";
28 | return 0;
29 |}
30
```

	Input	Expected	Got	
~	5	0	0	~
	7 10 -2 12 -1			
~	10	Not found	Not	~
	-12 -5 -4 -12 -14 5 10 -8 4 -3		found	
~	20	17 16	17 16	~
	0 4 8 2 4 8 6 1 3 2 8 5 1 6 5 4 7 7 2 0			
~	30	23 14 11 4	23 14	~
	7 8 2 5 7 2 4 9 9 6 3 7 9 3 7 5 8 2 0 8 1 0 2 7 9 2 4 0 4 8	0	11 4 0	
~	100	Not found	Not	~
	-79 38 50 89 14 -27 -85 85 -57 -77 -6 74 37 -23 16 24 -27 44 68 -27 83 -28 49 71 -46 -13		found	
	-98 -1 -29 50 -26 46 29 -91 84 -84 -8 -60 -35 77 -29 -78 -72 -90 61 37 85 -35 77 -16 -4 2			
	-72 -55 -87 31 71 75 95 22 99 -98 -93 -79 45 24 93 64 -79 -53 -35 35 -50 54 28 14 81 -42			
	-93 40 -19 -56 33 -43 81 22 70 21 52 -4 59 -70 44 -82 89 -52 4 -84 30 -67			

Correct

Marks for this submission: 10.00/10.00.

Back to Course