

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  $n$  ( $n \leq 1000$ );
- 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 trí yêu cầu.

Phân tách các phần 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	

Answer: (penalty regime: 0 %)

```

1  #include <bits/stdc++.h>
2  #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;
9      vector<int> vt;
10     while(n--)
11     {
12         int x; cin >> x;
13         vt.push_back(x);
14     }
15     int x, y; cin >> x >> y;
16     vt.insert(vt.begin()+y-1,x);
17     for(auto &x : vt) cout << x << " ";
18     return 0;
19 }
20

```

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

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.



## 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  $n$  ( $n \leq 1000$ );
- 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	

**Answer:** (penalty regime: 0 %)

```

1  #include <bits/stdc++.h>
2  #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;
9      vector<int> vt;
10     while(n--)
11     {
12         int x; cin >> x;
13         vt.push_back(x);
14     }
15     int y; cin >> y;
16     vt.erase(vt.begin()+y-1);
17     for(auto &x : vt) cout << x << " ";
18     return 0;
19 }
20
```

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

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.



## 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  $n$  ( $n < 100$ );
- 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 5 7 2 6 4	2 4 5 6 7

Answer: (penalty regime: 0 %)

```
1 #include <bits/stdc++.h>
2 #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;
9     vector<int> vt;
10    while(n--)
11    {
12        int x; cin >> x;
13        vt.push_back(x);
14    }
15    sort(vt.begin(),vt.end());
16    for(auto &x : vt) cout << x << " ";
17    return 0;
18 }
19
```

	Input	Expected	Got	
✓	5 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	✓

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.

## 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  $n$  ( $n < 100$ );
- 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	

Answer: (penalty regime: 0 %)

```
1 #include <bits/stdc++.h>
2 #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;
9     vector<int> vt;
10    while(n--)
11    {
12        int x; cin >> x;
13        vt.push_back(x);
14    }
15    sort(vt.begin(),vt.end(),greater<int>());
16    for(auto &x : vt) cout << x << " ";
17    return 0;
18 }
19
```



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

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.

## 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  $n$  ( $n \leq 100$ );
- 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	

**Answer:** (penalty regime: 0 %)

```

1  #include <bits/stdc++.h>
2  #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;
9      vector<int> vtc,vtl;
10     while(n-->0)
11     {
12         int x; cin >> x;
13         x % 2 ? vtl.push_back(x) : vtc.push_back(x);
14     }
15     sort(vtc.begin(),vtc.end());
16     sort(vtl.begin(),vtl.end(),greater<int>());
17     for(auto &x : vtc) cout << x <<" ";
18     for(auto &x : vtl) cout << x <<" ";
19     return 0;
20 }
21
```

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

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.



## 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	

Answer: (penalty regime: 0 %)

```

1  #include <bits/stdc++.h>
2  #define el "\n"
3  #define fi firrt
4  #define se second
5  using namespace std;
6  int main() {
7      ios_base::sync_with_stdio(0);
8      cin.tie(0);
9      cout.tie(0);
10     string s1, s2, s3, s4;
11     //cin >> s1 >> s2 >> s3 >> s4;
12     getline(cin, s1);
13     getline(cin, s2);
14     getline(cin, s3);
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 << " ";
23     return 0;
24 }
25

```

	Input	Expected	Got	
✓	May Troì Cua Em	C D A B	C D A B	✓
✓	Hoa hong hoa hue hoa lan hoa huong duong	A B D C	A B D C	✓
✓	Rau den rau day Rau muong ca rot	D A C B	D A C B	✓
✓	Thu hai Thu ba Chu nhat Thu bay	C A B D	C A B D	✓
✓	1 3 0 2	C A D B	C A D B	✓

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.

**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  $(n \setminus)$  số lượng khách mời trong danh sách trên một dòng,  $(n \setminus)$  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 Anastatia Alice Ariel Aurora Nemo	Aurora and Anastatia

**Answer:** (penalty regime: 0 %)

```

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

```

	Input	Expected	Got	
✓	5 Anastatia Alice Ariel Aurora Nemo	Aurora and Anastatia	Aurora and Anastatia	✓
✓	10 Big Al Belle Bert Nemo Aurora Alice Beast Anastatia Ariel Baloo	Bert and Aurora	Bert and Aurora	✓
✓	20 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	Baloo and Beast	Baloo and Beast	✓

	Input	Expected	Got	
✓	30 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	Captain Hook and Esmeralda	Captain Hook and Esmeralda	✓



	Input	Expected	Got	
✓	50 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	Green Army Men and Goofy	Green Army Men and Goofy	✓

	Input	Expected	Got	
	Daisy Duck Fairy Godmother			

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.

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. fire	1

Answer: (penalty regime: 0 %)

```
1 #include <iostream>
2 #include <string>
3
4 using namespace std;
5
6 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()) {
13        count++;
14        pos = sentence.find(keyword, pos + keyword.length());
15    }
16
17    cout << count << endl;
18
19    return 0;
20 }
21
```

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

	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. fate	0	0	✓
✓	And ignite. Ignite. Ignite. Ignite. Ignite	3	3	✓

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.

## 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:**

Input	Result
5 -1.61996539202 0.968616409208 2.12587576231 7.56980530885 11.305029863 -2.50635869654	-2.51 -1.62 0.97 2.13 7.57 11.31

**Answer:** (penalty regime: 0 %)

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

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

Marks for this submission: 10.00/10.00.

## 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  $\lfloor \frac{1}{2} \rfloor$ .

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: 1 3 2 2 5 2 3 7

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

For example:

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

Answer: (penalty regime: 0 %)

```

1  #include <bits/stdc++.h>
2  using namespace std;
3  #define ll long long
4  #define ull unsigned long long
5  #define el "\n"
6  const int mod=1e9+7;
7
8  int findLHS(int a[], int n)
9  {
10     int ans = 1;
11     map<int,int>mp;
12     vector<int>vt;
13     for(int i = 0; i < n; i++)
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++)
21     {
22         ans = max(ans, mp[vt[i]]);
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
33 void print()
34 {
35     int n; cin >> n;
36     int a[n];
37     for(auto &x : a) cin >> x;
38     cout << findLHS(a, n);
39 }
40

```



	Test	Input	Expected	Got	
✓	findLHS(arr, n)	4 1 2 3 4	2	2	✓
✓	findLHS(arr, n)	25 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	2	2	✓
✓	findLHS(arr, n)	50 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	6	6	✓
✓	findLHS(arr, n)	150 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	6	6	✓
✓	findLHS(arr, n)	250 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	15	15	✓

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.

## 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 * * *
* * * Q *
Q * * * *
* * Q * *

```

For example:

Input	Result
4 3 2 1 0	<pre> * * * * Q * * * Q * * * Q * * * Q * * * Q * * * * </pre>

Answer: (penalty regime: 0 %)

```

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

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.



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

Answer: (penalty regime: 0 %)

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

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

	Input	Expected	Got	
✓	5 7 10 -2 12 -1	0	0	✓
✓	10 -12 -5 -4 -12 -14 5 10 -8 4 -3	Not found	Not found	✓
✓	20 0 4 8 2 4 8 6 1 3 2 8 5 1 6 5 4 7 7 2 0	17 16	17 16	✓
✓	30 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	23 14 11 4 0	23 14 11 4 0	✓
✓	100 -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 -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	Not found	Not found	✓

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.

[Back to Course](#)