

Question 1

Not answered

Mark 0.00 out of 10.00

[ArrayBegins]

Mảng là một tập hợp các phần tử cố định có cùng một kiểu, được lưu trữ liên tiếp nhau trong các ô nhớ. Kiểu phần tử có thể là các kiểu bất kỳ như: ký tự, số, chuỗi ký tự...; cũng có khi ta sử dụng kiểu mảng để làm kiểu phần tử cho một mảng (trong trường hợp này ta gọi là mảng của mảng hay mảng nhiều chiều).

Mảng là kiểu dữ liệu được sử dụng rất thường xuyên. Chẳng hạn người ta cần quản lý một danh sách họ và tên của khoảng 100 sinh viên trong một lớp. Nhận thấy rằng mỗi họ và tên để lưu trữ ta cần một biến kiểu chuỗi, như vậy 100 họ và tên thì cần khai báo 100 biến kiểu chuỗi. Nếu khai báo như thế này thì đoạn khai báo cũng như các thao tác trên các họ tên sẽ rất dài dòng và rắc rối. Vì thế, kiểu dữ liệu mảng giúp ích ta trong trường hợp này; chỉ cần khai báo một biến, biến này có thể coi như là tương đương với 100 biến chuỗi ký tự; đó là 1 mảng mà các phần tử của nó là chuỗi ký tự. Hay như để lưu trữ các từ khóa của ngôn ngữ lập trình C, ta cũng dùng đến một mảng để lưu trữ chúng.

Viết một chương trình nhận vào từ bàn phím số nguyên n ($n \leq 20$), và n số nguyên. Lưu trữ các số nguyên này vào một mảng, sau đó in các giá trị của mảng đó ra màn hình, mỗi giá trị cách nhau một dấu cách.

For example:

Input	Result
3 1 2 3	1 2 3

Answer: (penalty regime: 0 %)

1	
---	--

Question 2

Not answered

Mark 0.00 out of 10.00

[PrintArray]

Viết hàm **void readArray(int a[], int n)** để đọc vào n số nguyên từ bàn phím và lưu vào mảng số nguyên a và hàm **void printArray(int a[], int n)** để in các phần tử của mảng a có số lượng phần tử là n , các phần tử cách nhau bởi dấu cách.

Lưu ý: chỉ cần viết 2 hàm như đề bài yêu cầu, không cần viết hàm **main()**, không cần viết các câu lệnh **#include**, **using namespace std ...**

For example:

Test	Input	Result
<pre>int n = 5; int a[n]; readArray(a, n); printArray(a, n);</pre>	1 2 3 4 5	1 2 3 4 5

Answer: (penalty regime: 0 %)

1	
---	--

Question 3

Not answered

Mark 0.00 out of 10.00

[SumOfArray]

Viết chương trình tính tổng các phần tử trong một mảng gồm n số nguyên.

Đầu vào

Đầu vào từ bàn phím gồm hai dòng:

- Dòng đầu tiên chứa số nguyên n ($n \leq 100$);
- 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 tổng các phần tử trong mảng.

For example:

Input	Result
3 1 2 3	6

Answer: (penalty regime: 0 %)

1	
---	--

Question 4

Not answered

Mark 0.00 out of 10.00

[Statistics 1]

Viết chương trình tính giá trị trung bình và phương sai của một dãy số thực.

Đầu vào

Đầu vào từ bàn phím gồm hai dòng:

- Dòng đầu tiên chứa số nguyên n ($n \leq 100$) là số lượng phần tử của dãy số;
- Dòng thứ hai chứa n số thực là các phần tử của dãy số đó. Các số phân tách nhau bởi dấu cách.

Đầu ra

In ra màn hình giá trị trung bình và phương sai của dãy số (làm tròn đến 2 chữ số thập phân sau dấu phẩy).

Phân tách các đáp án trong kết quả đầu ra bằng **một dấu cách duy nhất**.

For example:

Input	Result
5 1 2 3 4 5	3.00 2.00

Answer: (penalty regime: 0 %)

1	
---	--

Question 5

Not answered

Mark 0.00 out of 10.00

[StockIndex]

Chỉ số thị trường chứng khoán là một giá trị [thống kê](#) phản ánh tình hình của thị trường cổ phiếu. Nó được [tổng hợp](#) từ danh mục các cổ phiếu theo phương pháp tính nhất định. Thông thường, danh mục sẽ bao gồm các cổ phiếu có những điểm chung như cùng niêm yết tại một sở giao dịch chứng khoán, cùng ngành hay cùng mức vốn hóa thị trường. Các chỉ số chứng khoán này có thể do sở giao dịch chứng khoán định ra (ví dụ Vn-Index), cũng có thể do hãng thông tin (ví dụ Nikkei 225) hay một thể chế tài chính nào đó định ra (ví dụ Hang Seng Index).

Chỉ số chứng khoán sẽ lên xuống mỗi ngày. Viết một chương trình nhận vào từ bàn phím số nguyên n ($n \leq 100$) và độ tăng giảm của chỉ số chứng khoán trong n ngày. In ra màn hình tất cả biến động của chỉ số chứng khoán trong hai ngày liên tiếp (giá trị tuyệt đối của chênh lệch chỉ số chứng khoán trong hai ngày liên tiếp).

(Các con số được làm tròn tới hai chữ số thập phân sau dấu phẩy.)

For example:

Input	Result
5 -4.87 0.75 -0.37 -3.32 -4.94	5.62 1.12 2.95 1.62

Answer: (penalty regime: 0 %)

1	
---	--

Question 6

Not answered

Mark 0.00 out of 10.00

[Characters]

Biểu diễn [xâu kí tự](#) bằng mảng kiểu char

Trong các biểu diễn này, mỗi [xâu kí tự](#) là một [mảng một chiều](#) gồm các phân tử kiểu **char** và và thúc bởi ký tự **null '\0'**. Ví dụ, một [xâu kí tự](#) có thể được khai báo bằng một trong hai cách như sau như sau

```
char xau[6] = {'H', 'e', 'l', 'l', 'o', '\0'};
char xau[] = "Hello";
```

Ngôn ngữ C++ cung cấp một số hàm để xử lý trên [xâu kí tự](#) như sau

- **strcpy(s1, s2)**; Sao chép chuỗi s2 cho chuỗi s1.
- **strcat(s1, s2)**; Nối chuỗi s2 vào cuối chuỗi s1.
- **strlen(s1)**; Trả về độ dài của chuỗi s1.
- **strcmp(s1, s2)**; Trả về 0 nếu s1 và s2 là như nhau; nhỏ hơn 0 nếu s1<s2; lớn hơn 0 nếu s1>s2.

Dưới đây là một ví dụ cho việc sử dụng các hàm trên:

```
#include
#include

using namespace std;

int main ()
{
    char str1[10] = "Hello";
    char str2[10] = "Christmas";
    char str3[10];
    int len ;

    // sao chép str1 vào trong str3
    strcpy( str3, str1);
    cout << "strcpy( str3, str1) : " << str3 << endl;

    // nối hai xâu: str1 và str2
    strcat( str1, str2);
    cout << "strcat( str1, str2): " << str1 << endl;

    // do dài của str1 một sau khi thực hiện nối xâu
    len = strlen(str1);
    cout << "strlen(str1): " << len << endl;

    return 0;
}
```

Chạy chương trình trên sẽ cho kết quả như sau:

```
strcpy( str3, str1) : Hello
strcat( str1, str2): HelloWorld
strlen(str1) : 10
```

Bài tập

Cho 2 [xâu kí tự](#) s và t có cùng độ dài, không quá 99 kí tự, không có dấu cách. Viết chương trình đếm số lượng vị trí i sao cho $s[i] = t[i]$.

Đầu vào

Dữ liệu vào từ bàn phím gồm 2 dòng. Dòng thứ nhất chứa xâu s . Dòng thứ hai chứa xâu t .

Đầu ra

In ra màn hình theo yêu cầu ở trên.

For example:

Input	Result
abc123	2
acb213	

Answer: (penalty regime: 0 %)

Reset answer

```
1 #include <iostream>
2 #include <cstring>
3
4 using namespace std;
5
6 int main () {
7
8     char s[100],t[100];
9     cin >> s;
10    cin >> t;
11    // code của bạn ở đây
12
13    return 0;
14 }
```

Question 7

Not answered

Mark 0.00 out of 10.00

[WordCounting]

Đơn vị cơ sở để cấu tạo từ tiếng Việt là các *tiếng*, cái mà ngữ âm học vẫn gọi là các *âm tiết*.

Trong trình soạn thảo văn bản quy định, các tiếng được cách nhau bởi một khoảng trắng (dấu cách).

Viết chương trình nhập vào một đoạn văn bản và in ra màn hình số tiếng có trong đoạn văn bản đó (các khoảng trắng với độ rộng bất kì được tính như một dấu cách).

Hint: Đoạn văn bản cách nhau bởi dấu cách, kết thúc nhập văn bản khi gặp dấu *Enter*

For example:

Input	Result
tiếng	1 word
Tin học cơ sở 4	5 words

Answer: (penalty regime: 0 %)

1	
---	--

Question 8

Not answered

Mark 0.00 out of 10.00

[TwoDimensionArray]

Ở [bài trước](#), các bạn đã được thực hành với [mảng một chiều](#) trên ngôn ngữ C++. Bây giờ các bạn thử tưởng tượng nếu kiểu dữ liệu của [mảng một chiều](#) là [mảng một chiều](#). Hay nói cách khác, chúng ta có một mảng chứa các [mảng một chiều](#). Lúc này, chúng trở thành [mảng hai chiều](#).

Cũng như [mảng một chiều](#), ta có thể khai báo [mảng hai chiều](#) bằng cách dùng mảng tĩnh hoặc vector như sau

```
// Mảng tĩnh
<tên kiểu phần tử> <tên mảng>ước [<số hàng>][ <số cột>;
```

```
// Vector
vector < vector < <tên kiểu phần tử> > > <tên mảng>(<số hàng>, vector < <tên kiểu phần tử> >(<số cột>));
```

Ví dụ, để lưu một ma trận số nguyên có kích thước 3×5 , ta khai báo như sau,

```
// Mảng tĩnh
int a[3][5];
```

```
// Vector
vector < vector<int> > a(3,vector<int>(5));
```

Các hàng và cột trong [mảng hai chiều](#) cũng được đánh số từ 0. Ví dụ, một [mảng hai chiều](#) 3×5 được tạo thành như sau.

	Cột1	Cột2	Cột3	Cột4	Cột5
Hàng1	[0][0]	[0][1]	[0][2]	[0][3]	[0][4]
Hàng2	[1][0]	[1][1]	[1][2]	[1][3]	[1][4]
Hàng3	[2][0]	[2][1]	[2][2]	[2][3]	[2][4]

Để đọc một [mảng hai chiều](#) 3×5 từ một đầu vào, ta có thể sử dụng hai [vòng lặp lồng nhau](#) như sau,

```
for (int i = 0; i < 3; ++i)
    for (int j = 0; j < 5; ++j)
        cin >> a[i][j];
```

Bài tập

Cho ma trận kích thước $m \times n$ chứa các số nguyên, các hàng được đánh số từ 1 đến m , các cột được đánh số từ 1 đến n . Có Q câu hỏi, mỗi câu hỏi đưa ra 2 số i, j . Nhiệm vụ của bạn là in ra giá trị của phần tử ở hàng thứ i và cột thứ j của ma trận.

Đầu vào

Đầu vào từ bàn phím gồm $m + Q + 1$ dòng.

- Dòng đầu tiên chứa 3 số nguyên m, n, Q ($m, n, Q \leq 500$).
- m dòng tiếp theo mỗi dòng chứa n số nguyên biểu diễn ma trận đã cho, các số liên tiếp trên một dòng cách nhau bởi một dấu cách.
- Q dòng tiếp theo, mỗi dòng là một câu hỏi chứa 2 số nguyên i, j cách nhau bởi một dấu cách.

Đầu ra

In ra màn hình Q dòng, mỗi dòng tương ứng với câu trả lời của một câu hỏi.

For example:

Input	Result
3 3 2	5
1 3 5	4
2 4 7	
3 4 2	
1 3	
2 2	

Answer: (penalty regime: 0 %)

1		
---	--	--

Question 9

Not answered

Mark 0.00 out of 10.00

[MatrixQueries]

Cho ma trận kích thước $m \times n$ chứa các số nguyên, các hàng được đánh số từ 1 đến m , các cột được đánh số từ 1 đến n . Có Q câu hỏi, mỗi câu hỏi đưa ra 2 số i, j . Nhiệm vụ của bạn là in ra giá trị của phần tử ở hàng thứ i và cột thứ j của ma trận.

Đầu vào

Đầu vào từ bàn phím gồm $m + Q + 1$ dòng.

- Dòng đầu tiên chứa 3 số nguyên n, m, Q ($m, n, Q \leq 1000$).
- m dòng tiếp theo mỗi dòng chứa n số nguyên biểu diễn ma trận đã cho, các số liên tiếp trên một dòng cách nhau bởi một dấu cách.
- Q dòng tiếp theo, mỗi dòng là một câu hỏi chứa 2 số nguyên i, j cách nhau bởi một dấu cách.

Đầu ra

In ra màn hình Q dòng, mỗi dòng tương ứng với câu trả lời của một câu hỏi.

For example:

Input	Result
3 3 2	5
1 3 5	4
2 4 7	
3 4 2	
1 3	
2 2	

Answer: (penalty regime: 0 %)

1	
---	--

Question 10

Not answered

Mark 0.00 out of 10.00

[MatrixAddition]

Cho đầu vào là 2 ma trận A, B có cùng kích thước ($m \times n$).

Viết chương trình tính tổng của 2 ma trận trên và in kết quả ra màn hình.

Kết quả làm tròn đến chữ số thập phân thứ hai.

For example:

Input	Result
2 3	-246.76 -75.32 53.80
-211.478 -11.465 -46.896	-239.96 -201.85 196.49
-195.496 -42.879 12.055	
-35.2794 -63.858 100.6966	
-44.4654 -158.9674 184.4364	

Answer: (penalty regime: 0 %)

1	
---	--

[Back to Course](#)