

---How to get input in vector?

```
#include <iostream>
#include <vector>
using namespace std;


int main() {
    int n; // ভেক্টরের সাইজ
    cout << "Enter the number of elements: ";
    cin >> n;

    vector<int> vec(n); // ভেক্টর ডিক্লেয়ার করলাম

    cout << "Enter the elements:\n";
    for (int i = 0; i < n; ++i) {
        cin >> vec[i]; // ইনপুট নিচ্ছি
    }

    // ভেক্টরের এলিমেন্টগুলো প্রিন্ট করা
    cout << "The elements in the vector are:\n";
    for (int i = 0; i < n; ++i) {
        cout << vec[i] << " ";
    }
    cout << endl;

    return 0;
}
```

 Copy code

---vector sort code?

```
#include <algorithm> // sort() ফাংশনের জন্য

using namespace std;

int main() {
    int n;
    cout << "Enter the number of elements: ";
    cin >> n;

    vector<int> vec(n); // ভেক্টর ডিক্লেয়ার

    cout << "Enter the elements:\n";
    for (int i = 0; i < n; ++i) {
        cin >> vec[i]; // ইনপুট নিচ্ছি
    }

    // ভেক্টর সর্ট করা
    sort(vec.begin(), vec.end());

    cout << "Sorted vector in ascending order:\n";
    for (int i = 0; i < n; ++i) {
        cout << vec[i] << " ";
    }
    cout << endl;

    // Descending order এ সর্ট করতে:
    sort(vec.rbegin(), vec.rend());
    cout << "Sorted vector in descending order:\n";
    for (int i = 0; i < n; ++i) {
        cout << vec[i] << " ";
    }
    cout << endl;
```

❓ **sort(vec.begin(), vec.end()):**

- ভেক্টরের begin() থেকে end() পর্যন্ত সর্ট করে।
- ডিফল্ট ভাবে এটি আসেন্ডিং অর্ডারে সর্ট করে।

❓ **sort(vec.rbegin(), vec.rend()):**

রিভার্স ইটারেটর ব্যবহার করে ডেসেন্ডিং অর্ডারে স্ট করে।

---without int n but direct input vector code(with Ctrl+z):

For int:

```
cpp Copy code

#include <iostream>
#include <vector>
#include <algorithm> // sort() ফাংশনের জন্য
#include <iterator> // istream_iterator এর জন্য

using namespace std;

int main() {
    cout << "Enter the elements (press Ctrl+D or Ctrl+Z to stop):\n";

    vector<int> vec((istream_iterator<int>(cin)), istream_iterator<int>());

    // স্ট করা ভেক্টর
    sort(vec.begin(), vec.end());

    cout << "Sorted vector in ascending order:\n";
    for (int num : vec) {
        cout << num << " ";
    }
    cout << endl;

    return 0;
}
```

1. `istream_iterator<int>(cin)`:
 - এটি ইনপুট স্ট্রিম থেকে এলিমেন্টগুলো ভেক্টরে স্টোর করতে ব্যবহার করা হয়।
 - ইনপুট শেষ করার জন্য Ctrl+D (Linux/Mac) বা Ctrl+Z (Windows) চাপুন।
2. `sort(vec.begin(), vec.end())`: ভেক্টরের সব এলিমেন্ট স্ট করে।
3. **রেঞ্জ-বেসড লুপ**: স্ট করা ভেক্টরের এলিমেন্ট প্রিন্ট করতে ব্যবহার করা হয়েছে।

For Char:

```
#include <iostream>
#include <vector>
#include <iterator>
using namespace std;
```

```
int main() {
    vector<char> v;

    // istream_iterator দিয়ে char ইনপুট নেয়া
    cout << "Enter characters (press Enter twice to stop):" << endl;
    istream_iterator<char> it(cin); // ইনপুট স্ট্রিম থেকে চর একে একে পড়বে
    istream_iterator<char> eos;      // ইনপুট শেষ হলে ব্যবহার হবে
```

```
    // ইনপুট নেওয়া এবং ভেক্টরে যোগ করা
    while (it != eos) {
        v.push_back(*it);
        ++it;
    }
}
```

```

// Vector এর সব char প্রিন্ট করা
cout << "Entered characters: ";
for (char c : v) {
    cout << c << " ";
}
cout << endl;


return 0;
}

```

ব্যাখ্যা:

- `istream_iterator<char> it(cin);` - এটি `cin` স্ট্রিম থেকে `char` ইনপুট নেওয়ার জন্য একটি ইটারেটর তৈরি করে।
- `istream_iterator<char> eos;` - এটি ইনপুটের শেষ চিহ্ন হিসেবে কাজ করবে।
- `while (it != eos)` - যখন ইনপুট শেষ না হয়, তখন এটি চর একে একে ভেক্টরে যোগ করবে।

---without int n but direct input vector code(without Ctrl+z):

 Copy code

```
#include <iostream>
#include <vector>
#include <sstream> // stringstream এর জন্য
#include <algorithm> // sort() ফাংশনের জন্য

using namespace std;

int main() {
    cout << "Enter the elements in a single line:\n";

    string input;
    getline(cin, input); // এক লাইনে ইনপুট নেওয়া

    stringstream ss(input); // string থেকে স্ট্রিম তৈরি
    vector<int> vec;
    int num;

    while (ss >> num) { // একে একে সংখ্যাগুলো ভেক্টরে যোগ করা
        vec.push_back(num);
    }

    // ভেক্টর সর্ট করা
    sort(vec.begin(), vec.end());

    cout << "Sorted vector in ascending order:\n";
    for (int num : vec) {
        cout << num << " ";
    }
    cout << endl;

    return 0;
}
```



1. `getline(cin, input):`
 - পুরো একটি লাইন ইনপুট হিসেবে নেওয়া হয় (যেখানে ইনপুট আলাদা করতে স্পেস ব্যবহৃত হয়)।
 2. `stringstream ss(input):`
 - ইনপুট লাইনের প্রতিটি সংখ্যাকে পার্স (parse) করতে সাহায্য করে।
 3. `vec.push_back(num):`
 - প্রতিটি সংখ্যা ভেক্টরে যোগ করা হয়।
 4. `sort(vec.begin(), vec.end()):`
 - ভেক্টরের সংখ্যাগুলো সর্ট করা হয়।
-

---how to transfer input array to vector ?

১. সরাসরি কনস্ট্রাক্টর ব্যবহার করে

আপনার যদি ইনপুট অ্যারেটি আগে থেকেই থাকে, তাহলে `std::vector` সরাসরি অ্যারেটি থেকে কপি তৈরি করতে পারে:

```
cpp Copy code

#include <iostream>
#include <vector>

int main() {
    int arr[] = {1, 2, 3, 4, 5};
    int n = sizeof(arr) / sizeof(arr[0]); // অ্যারের সাইজ বের করা
```

```

// অ্যারে থেকে ভেক্টরে ট্রান্সফার
std::vector<int> vec(arr, arr + n);


// আউটপুট চেক
for (int x : vec) {
    std::cout << x << " ";
}
return 0;
}

```

২. std::copy ব্যবহার করে

std::copy ফাংশন ব্যবহার করেও অ্যারে থেকে ভেক্টরে ডেটা ট্রান্সফার করা যায়।

cpp

 Copy code

```

#include <iostream>
#include <vector>
#include <algorithm> // std::copy এর জন্য

int main() {
    int arr[] = {1, 2, 3, 4, 5};
    int n = sizeof(arr) / sizeof(arr[0]);

    std::vector<int> vec(n); // ভেক্টরের সাইজ প্রি-ডিফাইন করুন

    // std::copy দিয়ে ট্রান্সফার
    std::copy(arr, arr + n, vec.begin());

    for (int x : vec) {
        std::cout << x << " ";
    }
    return 0;
}

```


৩. ম্যানুয়ালি for লুপ ব্যবহার করে

এটি সাধারণত তখন ব্যবহার করা হয়, যখন অ্যারের কন্ডিশন বা কাস্টম প্রসেসিং করতে হয়।

```
cpp Copy code  
  
#include <iostream>  
#include <vector>  
  
int main() {  
    int arr[] = {1, 2, 3, 4, 5};  
    int n = sizeof(arr) / sizeof(arr[0]);  
  
    std::vector<int> vec;  
  
    // ম্যানুয়ালি পুশ ব্যাক  
    for (int i = 0; i < n; i++) {  
        vec.push_back(arr[i]);  
    }  
  
    for (int x : vec) {  
        std::cout << x << " ";  
    }  
    return 0;  
}
```

কোন পদ্ধতিটি বেছে নেবেন?

- **কনস্ট্রাক্টর:** যদি সরাসরি অ্যারে থেকে ভেক্টর তৈরি করতে চান।
- `std::copy`: যদি ইতিমধ্যে ভেক্টরের সাইজ নির্ধারণ করা থাকে।
- **for লুপ:** যদি কোনো কাস্টম প্রসেসিং দরকার হয়।


--- C++ e float number er decimal k 3 digit pojonto newa:

C++ তে float বা double নম্বরকে দশমিকের পরে ৩ ডিজিট পর্যন্ত প্রদর্শনের জন্য `iomanip` হেডার ফাইল ব্যবহার করা হয়। এর জন্য `std::fixed` এবং `std::setprecision` ম্যানিপুলেটর ব্যবহার করা হয়।

কীভাবে কাজ করে:

1. `std::fixed`: এইটা দশমিকের পরে নির্দিষ্ট সংখ্যক ডিজিট দেখাতে বলে।
2. `std::setprecision(3)`: এটি বলে যে দশমিকের পরে ৩ ডিজিট পর্যন্ত প্রদর্শন করতে।

cpp

 Copy code

```
#include <iostream>
#include <iomanip> // iomanip header file include করতে হবে

int main() {
    float number = 123.456789;

    // দশমিকের পরে ৩ ডিজিট দেখানোর জন্য
    std::cout << std::fixed << std::setprecision(3) << number << std::endl;

    return 0;
}
```

--- C++ e mixed(Capital & Small) word tranlatioin to uppcase & lowercase:

Lowercase:

1. `transform()` ফাংশন ব্যবহার করা হয়েছে: এটি স্ট্রিংয়ের সব অক্ষরকে ছোটো হাতের (lowercase) বা বড়ো হাতের (uppercase) বানানোর জন্য ব্যবহৃত হয়।

cpp

 Copy code

```
transform(s.begin(), s.end(), s.begin(), ::tolower);
```

এখানে `::tolower` ব্যবহার করা হয়েছে প্রতিটি অক্ষর ছোটো হাতের বানানোর জন্য।

```
#include<iostream>
#include<string>
#include<algorithm> // transform ফাংশনের জন্য
using namespace std;
```

```
int main()
{
    int n;
    cin >> n;
    while (n--) {
        string s;
        cin >> s;

        // পুরো স্ট্রিংকে ছোট অক্ষরে রূপান্তর করা
        transform(s.begin(), s.end(), s.begin(), ::tolower);

        if (s == "yes") {
            cout << "YES" << endl;
        } else {
            cout << "NO" << endl;
        }
    }
    return 0;
}
```

Uppercase:

As same as lowercase only change this line(code):

```
transform(s.begin(), s.end(), s.begin(), ::toupper);
```

