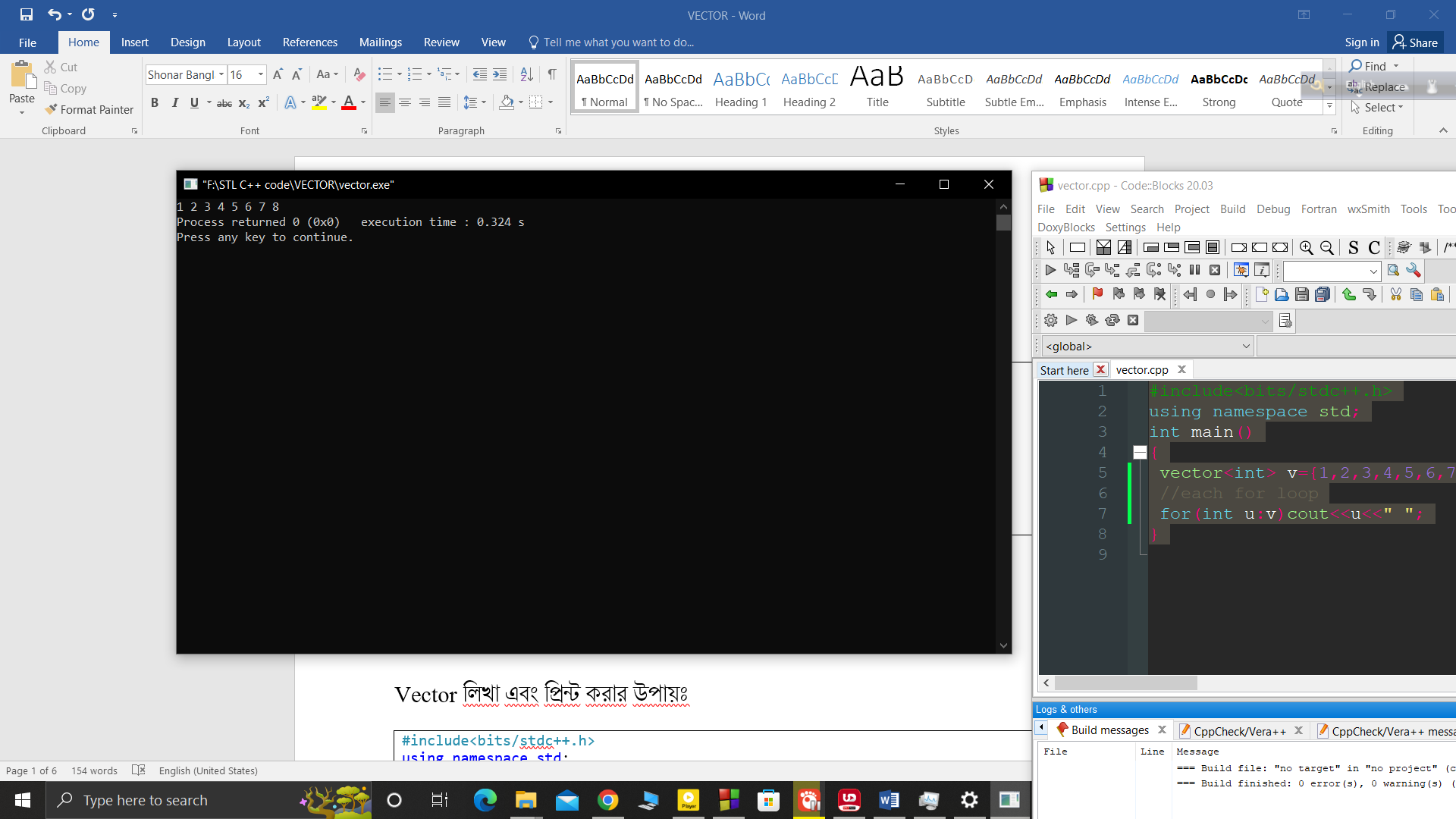
**VECTOR:**

Vector সহজে প্রিন্ট করার উপায়ঃ

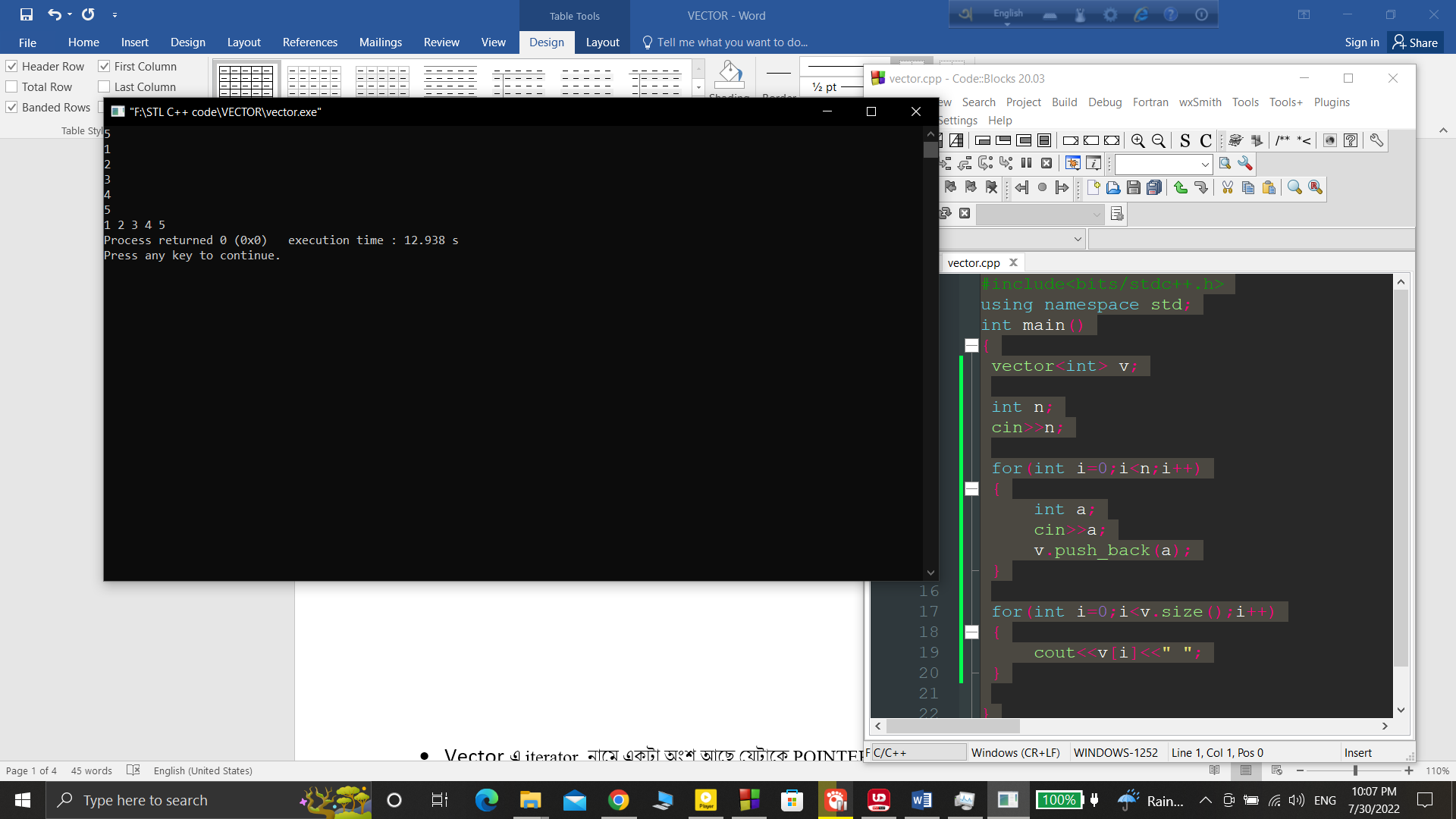
Each for loop usage:

|  |
| --- |
| #include<bits/stdc++.h> using namespace std; int main() {  vector<int> v={1,2,3,4,5,6,7,8};  //each for loop  for(int u:v)cout<<u<<" "; } |



Vector লিখা এবং প্রিন্ট করার উপায়ঃ

|  |
| --- |
| #include<bits/stdc++.h> using namespace std; int main() {  vector<int> v;    int n;  cin>>n;    for(int i=0;i<n;i++)  {      int a;      cin>>a;      v.push\_back(a);  }    for(int i=0;i<v.size();i++)  {      cout<<v[i]<<" ";  }  } |



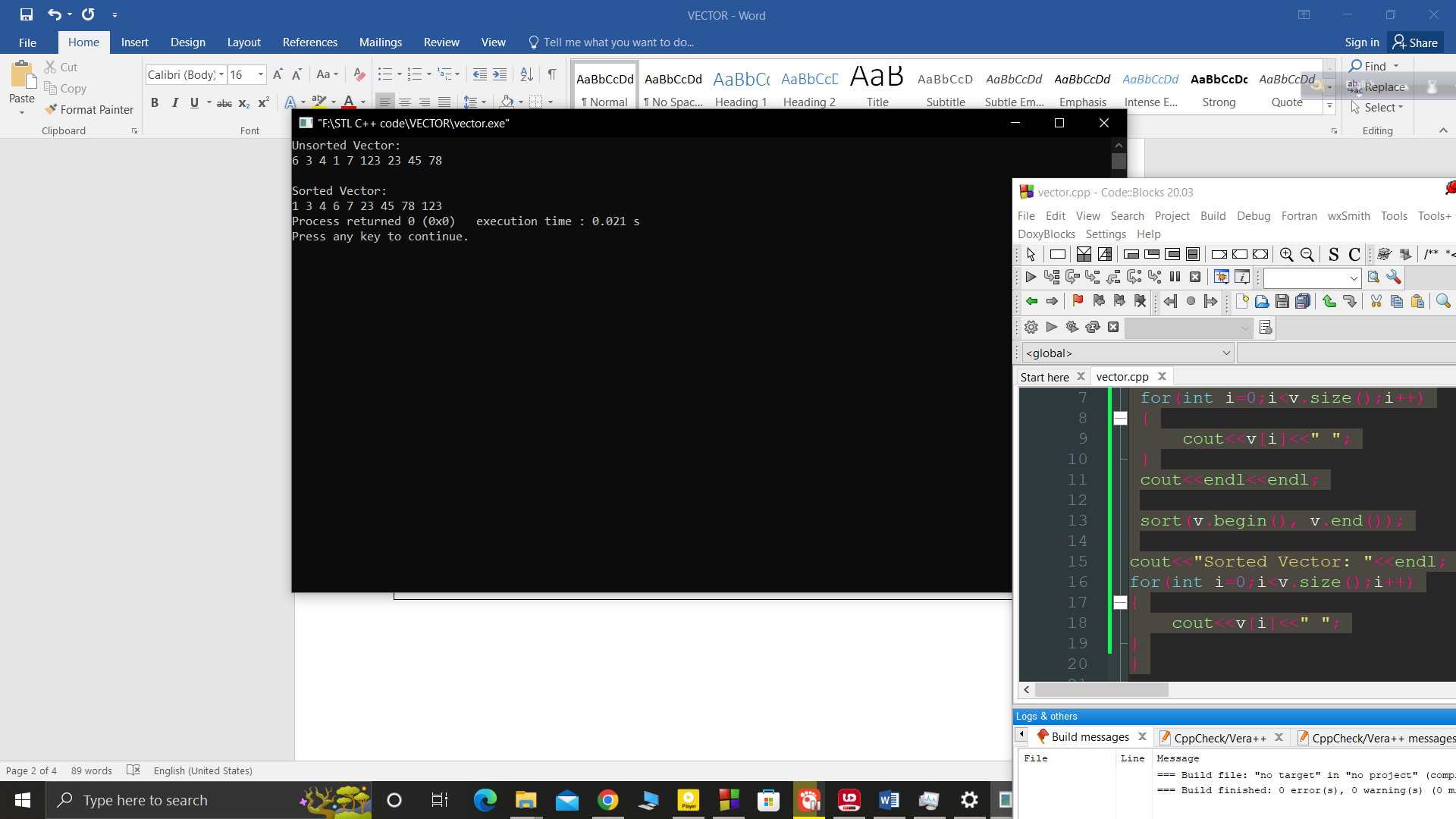
* push\_back(): পিছন থেকে element add করে;
* pop\_back() : পিছন থেকে element বাদ দেয়

sorting:

non-decreasing order :

sort(v.begin(), v.end());

|  |  |
| --- | --- |
| |  | | --- | | #include<bits/stdc++.h> using namespace std; int main() {  vector<int> v={6,3,4,1,7,123,23,45,78};  cout<<"Unsorted Vector:"<<endl;  for(int i=0;i<v.size();i++)  {      cout<<v[i]<<" ";  }  cout<<endl<<endl;   sort(v.begin(), v.end());  cout<<"Sorted Vector: "<<endl; for(int i=0;i<v.size();i++) {     cout<<v[i]<<" "; } } | |



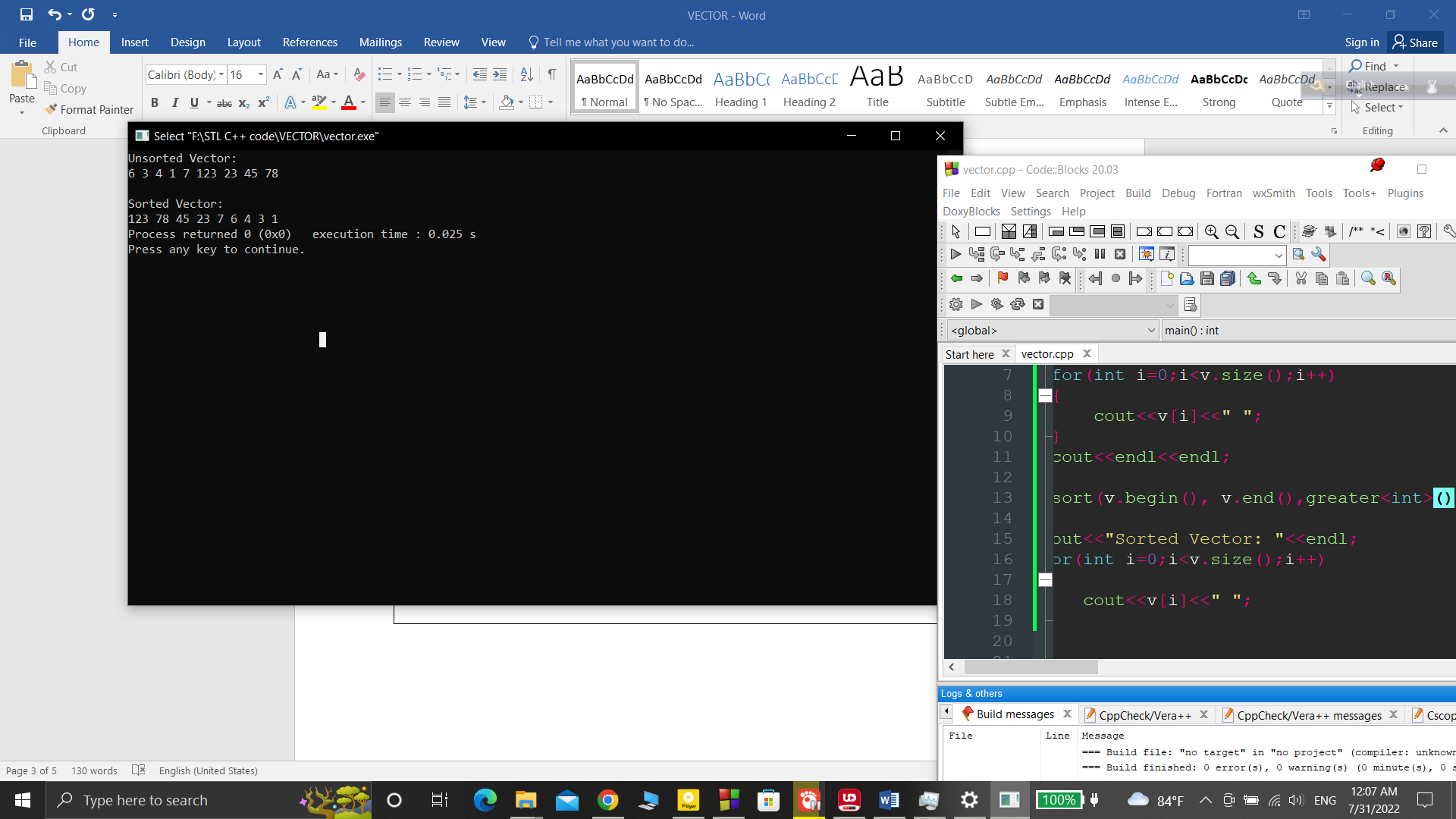
Non-increasing order:

sort(v.begin(), v.end(), greater<int>() ); //arrey র ক্ষেত্রেও use kora jabe

or

sort(v.rbegin(), v.rend());

|  |  |
| --- | --- |
| #include<bits/stdc++.h> using namespace std; int main() {  vector<int> v={6,3,4,1,7,123,23,45,78};  cout<<"Unsorted Vector:"<<endl;  for(int i=0;i<v.size();i++)  {      cout<<v[i]<<" ";  }  cout<<endl<<endl;   sort(v.rbegin(), v.rend());  cout<<"Sorted Vector: "<<endl; for(int i=0;i<v.size();i++) {     cout<<v[i]<<" "; } } |  |

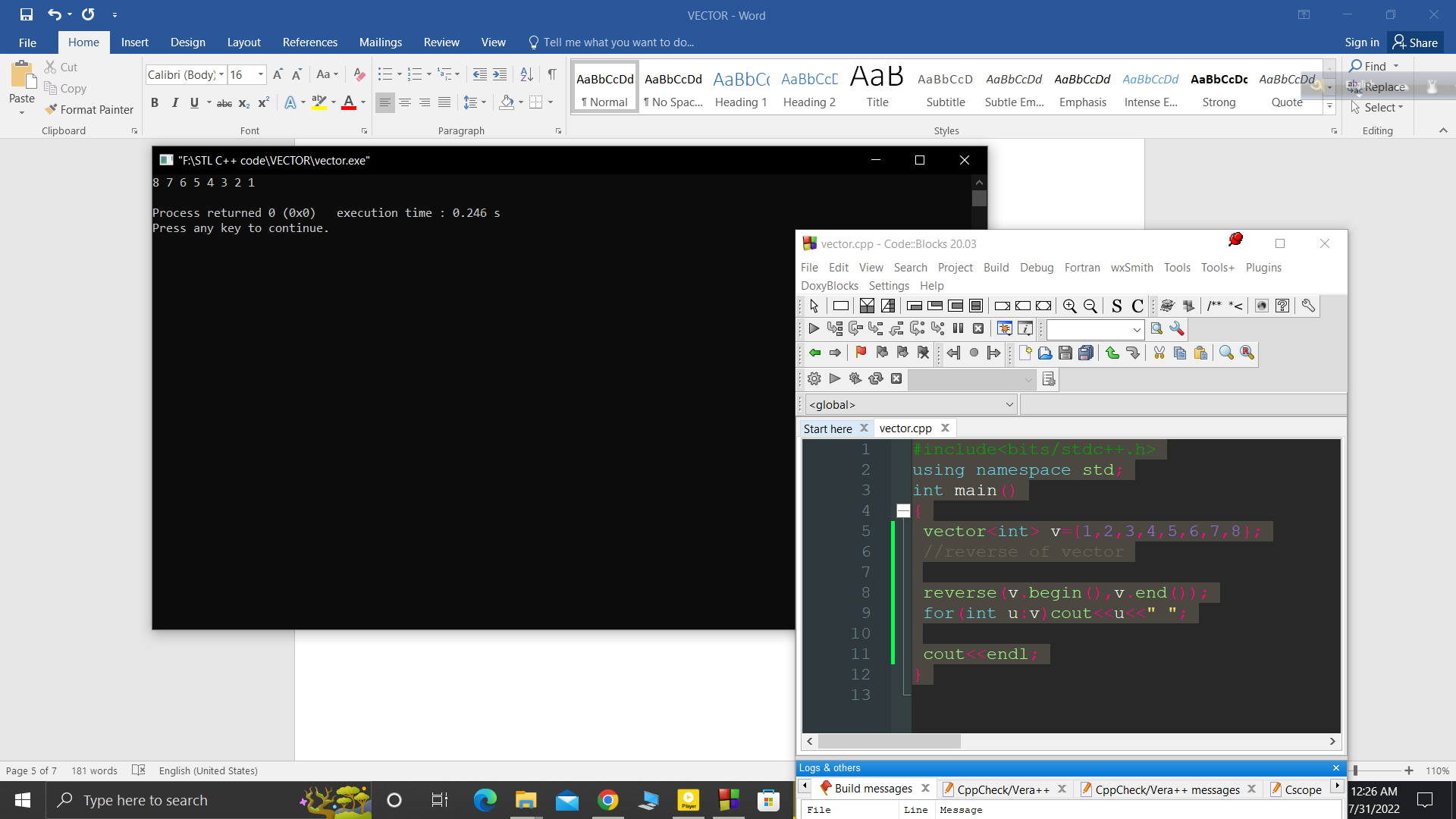


* Vector এ iterator নামে একটা অংশ আছে যেটাকে POINTER বলা হয় ।

Reverse of vector:

Reverse(v.begin(), v.end())

|  |  |
| --- | --- |
| |  | | --- | | #include<bits/stdc++.h> using namespace std; int main() {  vector<int> v={1,2,3,4,5,6,7,8};  //reverse of vector   reverse(v.begin(),v.end());  for(int u:v)cout<<u<<" ";    cout<<endl; } | |



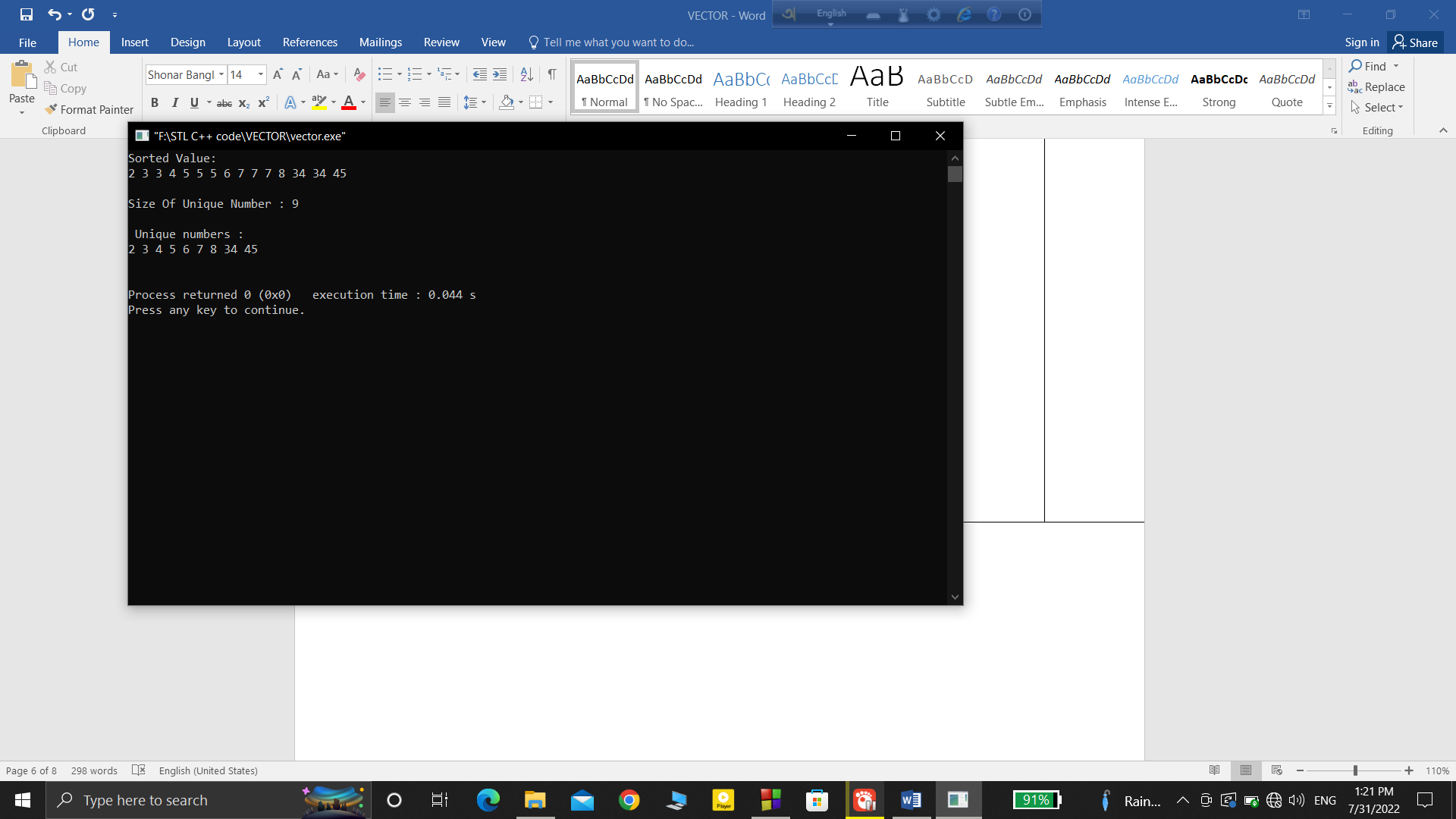
* V.back() : এর মাধ্যমে কোনো vector এর last element টাকে প্রিন্ট করা যায়;
* V.erase( v.begin( ) ): vector এর মধ্যে থাকা element গুলোর প্রথম element টা delete হয়ে গেছে।

Unique number বের করাঃ

প্রথমে sort করে নিতে হবে তারপর unique number এর size বের করতে হবে তারপর unique number গুলো প্রিন্ট করতে হবে ।

int sz = unique( v.begin( ) , v.end( ) ) – v.begin( ) ; //এর মাধ্যমে unique number এর size বের করা যায় ।

|  |  |  |
| --- | --- | --- |
| |  | | --- | | #include<bits/stdc++.h> using namespace std; int main() {  vector<int> v={2,3,3,4,5,5,5,6,7,7,7,8,34,34,45,};  //unique value return      //1st sort করে নিতে হবে  sort(v.begin(),v.end());   cout<<"Sorted Value: "<<endl;  for(auto u:v)cout<<u<<" ";    cout<<endl<<endl;      //2nd unique বের করতে হবে int sz = unique(v.begin(), v.end()) - v.begin();   cout<<"Size Of Unique Number : "<<sz<<endl<<endl;   cout<<" Unique numbers :  "<<endl;   for(int i=0;i<sz;i++) {  cout<<v[i]<<" "; }   cout<<endl<<endl; } | |  |



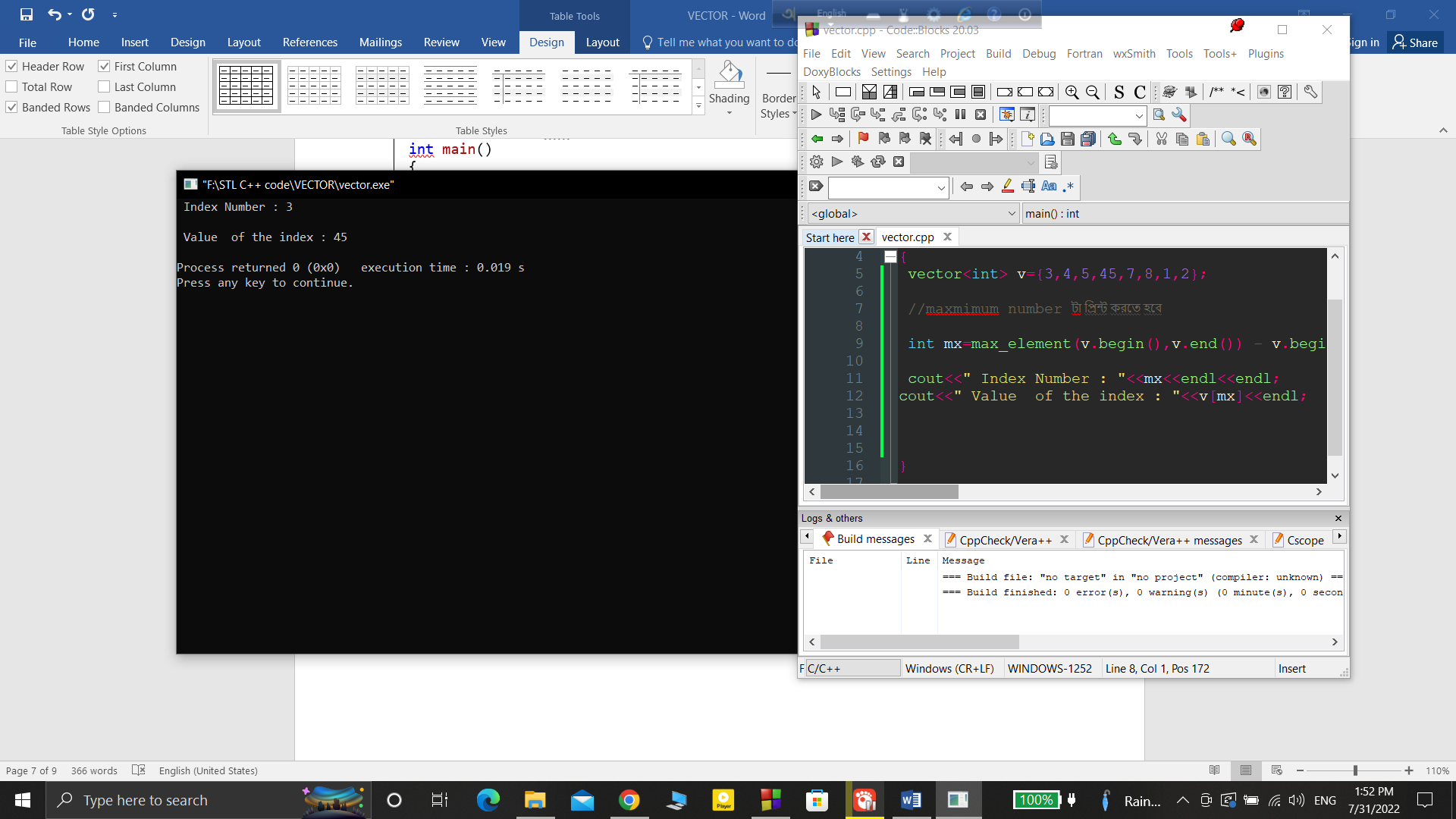
Vector এর মধ্যে Maximum element and Minimum element বের করার উপায় ঃ

ম্যাক্সিমাম বের করার উপায়ঃ

Index of the maximum number

int mx=max\_element(v.begin(),v.end()) - v.begin();

|  |  |
| --- | --- |
| |  | | --- | | #include<bits/stdc++.h> using namespace std; int main() {  vector<int> v={3,4,5,45,7,8,1,2};   //maxmimum number টা প্রিন্ট করতে হবে   int mx=max\_element(v.begin(),v.end()) - v.begin();   cout<<" Index Number : "<<mx<<endl<<endl;  cout<<" Value  of the index : "<<v[mx]<<endl; } | |



মিনিমাম নাম্বার বের করার উপায়ঃ

|  |
| --- |
| #include<bits/stdc++.h> using namespace std; int main() {  vector<int> v={3,4,5,45,7,8,1,2};   //maxmimum number টা প্রিন্ট করতে হবে   int mx=min\_element(v.begin(),v.end()) - v.begin();   cout<<" Index Number : "<<mx<<endl<<endl;  cout<<" Value  of the index : "<<v[mx]<<endl; } |

