

# Abstraction and Paradigms of Programming

# Taking User Input in Pair

```
Main(){  
    Pair<int,string>p;  
    Cin>>p.first;  
    Cin>>p.second;  
    Cout<<p.first<<endl;  
    Cout<<p.second<<endl;  
}
```

# Vectors in C++

- Vectors are same as dynamic arrays with the ability to resize itself automatically when an element is inserted or deleted.
- Their storage is handled automatically by the container.
- Vector elements are placed in contiguous storage so that they can be accessed and traversed using iterators. In vectors, data is inserted at the end.

# Creating Vectors

```
Main(){  
    vector<int>v;  
    int n;  
    Cin>>n;  
    For(int i=0;i<n;i++){  
        Int x;  
        Cin>>x;  
        v.push_back(x);  
    }  
}
```

# Printing the elements of a Vector

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

void PrintVec(vector<int> v){
    for(int i=0;i<v.size();i++){
        cout<<v[i]<<endl;
    }
}

int main(){
    vector<int> v;
    int n;
    cin>>n;
    for(int i=0;i<n;i++){
        int x;
        cin>>x;
        v.push_back(x);
    }
    PrintVec(v);
}
```

# Size() method to check the size of the vector

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

void PrintVec(vector<int> v){
    cout<<"size : "<<v.size()<<endl;
    for(int i=0;i<v.size();i++){
        cout<<v[i]<<endl;
    }
}

int main(){
    vector<int> v;
    int n;
    cin>>n;
    for(int i=0;i<n;i++){
        int x;
        cin>>x;
        v.push_back(x);
    }
    PrintVec(v);
}
```

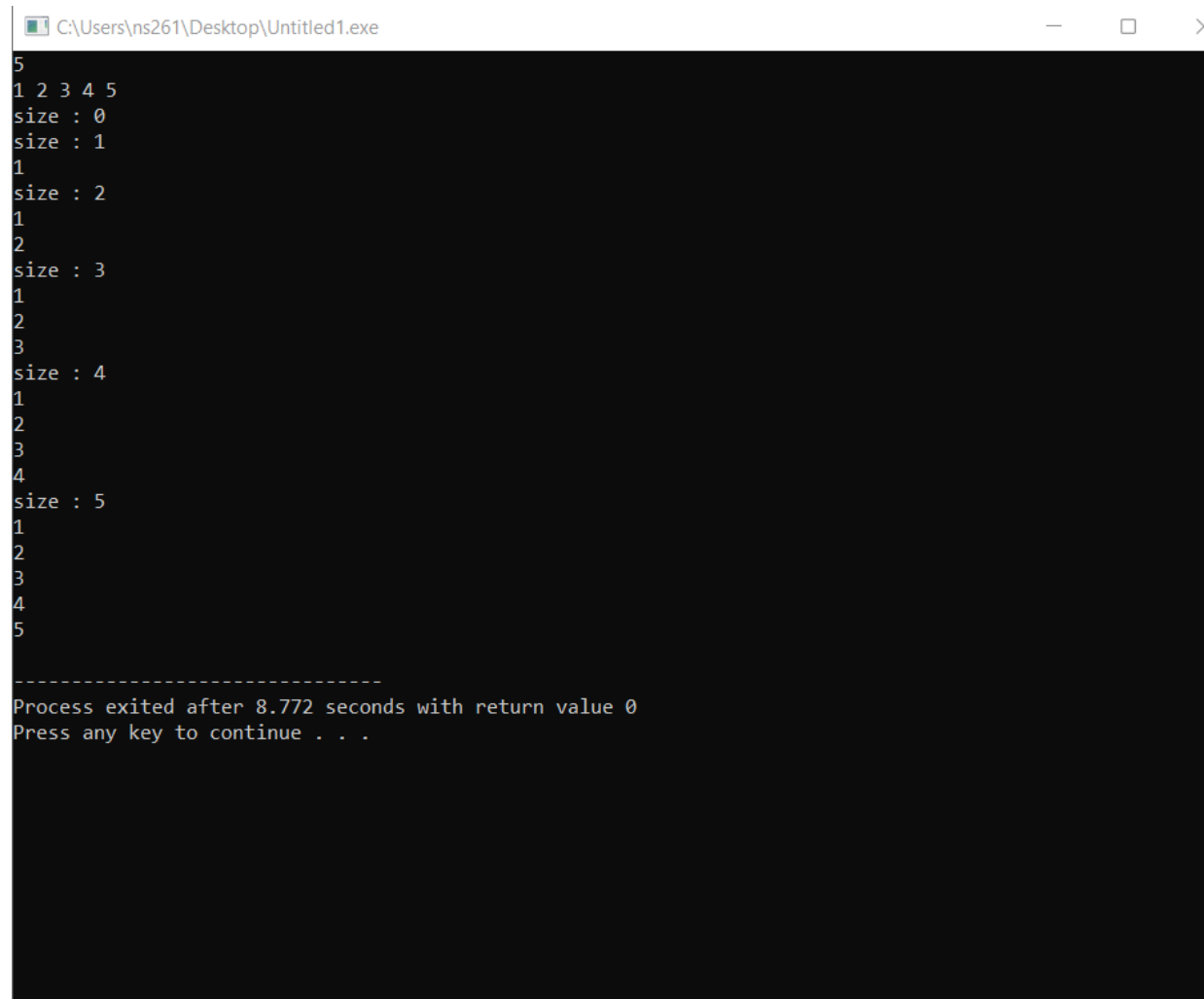
# Whether the Vector size is actually dynamic?

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

void PrintVec(vector<int> v){
    cout<<"size : "<<v.size()<<endl;
    for(int i=0;i<v.size();i++){
        cout<<v[i]<<endl;
    }
}

int main(){
    vector<int> v;
    int n;
    cin>>n;
    for(int i=0;i<n;i++){
        int x;
        cin>>x;
        PrintVec(v);
        v.push_back(x);
    }
    PrintVec(v);
}
```

# Output



```
C:\Users\ns261\Desktop\Untitled1.exe
5
1 2 3 4 5
size : 0
size : 1
1
size : 2
1
2
size : 3
1
2
3
size : 4
1
2
3
4
size : 5
1
2
3
4
5
-----
Process exited after 8.772 seconds with return value 0
Press any key to continue . . .
```



# Creating a Vector of Predefined Size

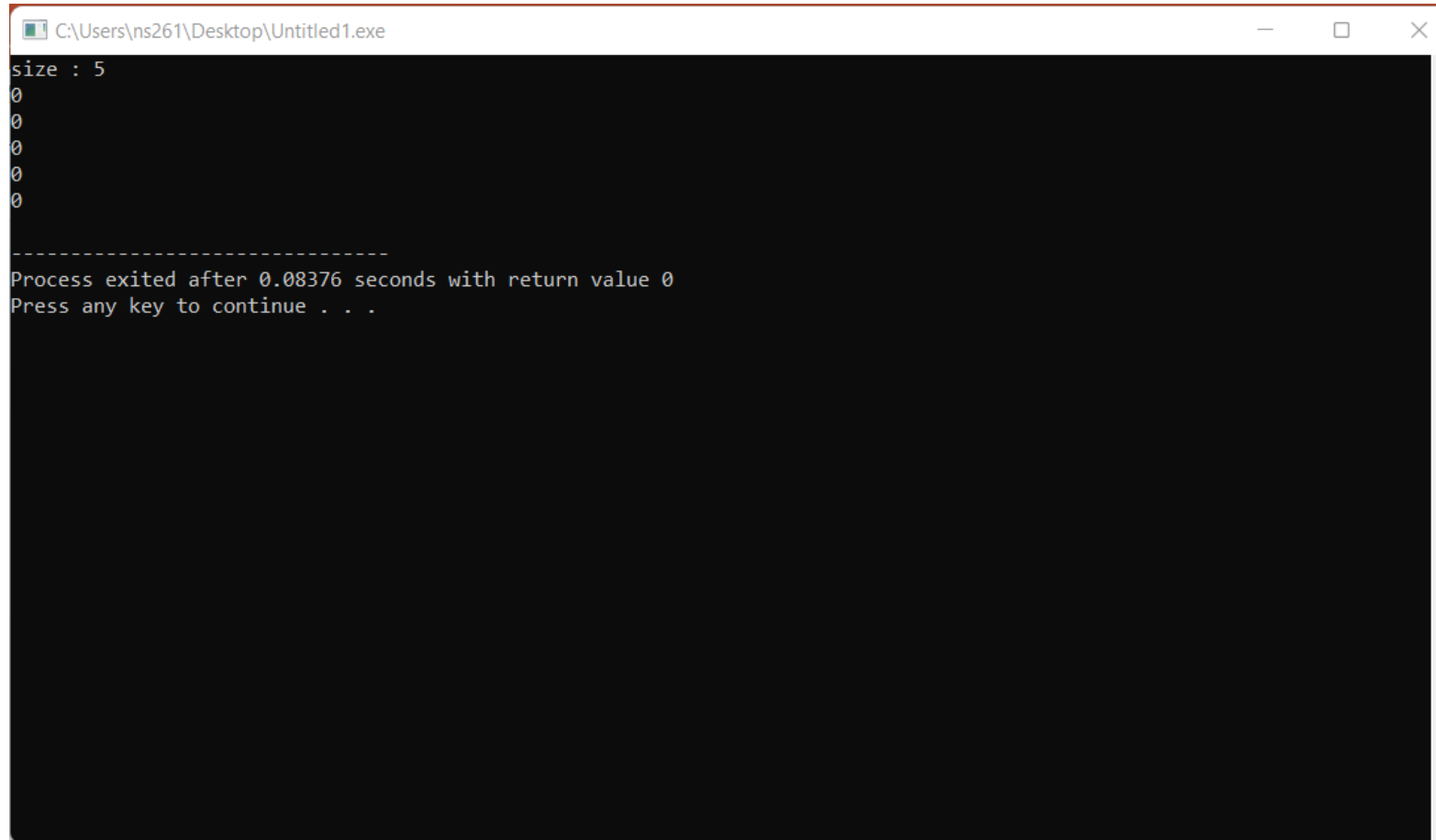
```
#include<iostream>
#include<vector>
using namespace std;

void PrintVec(vector<int> v){
    cout<<"size : "<<v.size()<<endl;
    for(int i=0;i<v.size();i++){
        cout<<v[i]<<endl;
    }
}

int main(){
    vector<int> v(5);
    //v.push_back(x);

    PrintVec(v);
}
```

# Output

A screenshot of a Windows command prompt window. The title bar at the top shows the file path "C:\Users\ns261\Desktop\Untitled1.exe" and standard window controls (minimize, maximize, close). The command prompt area has a black background with white text. The output of the program is as follows:

```
size : 5
0
0
0
0
0
0
-----
Process exited after 0.08376 seconds with return value 0
Press any key to continue . . .
```

# Can the predefined size be altered?

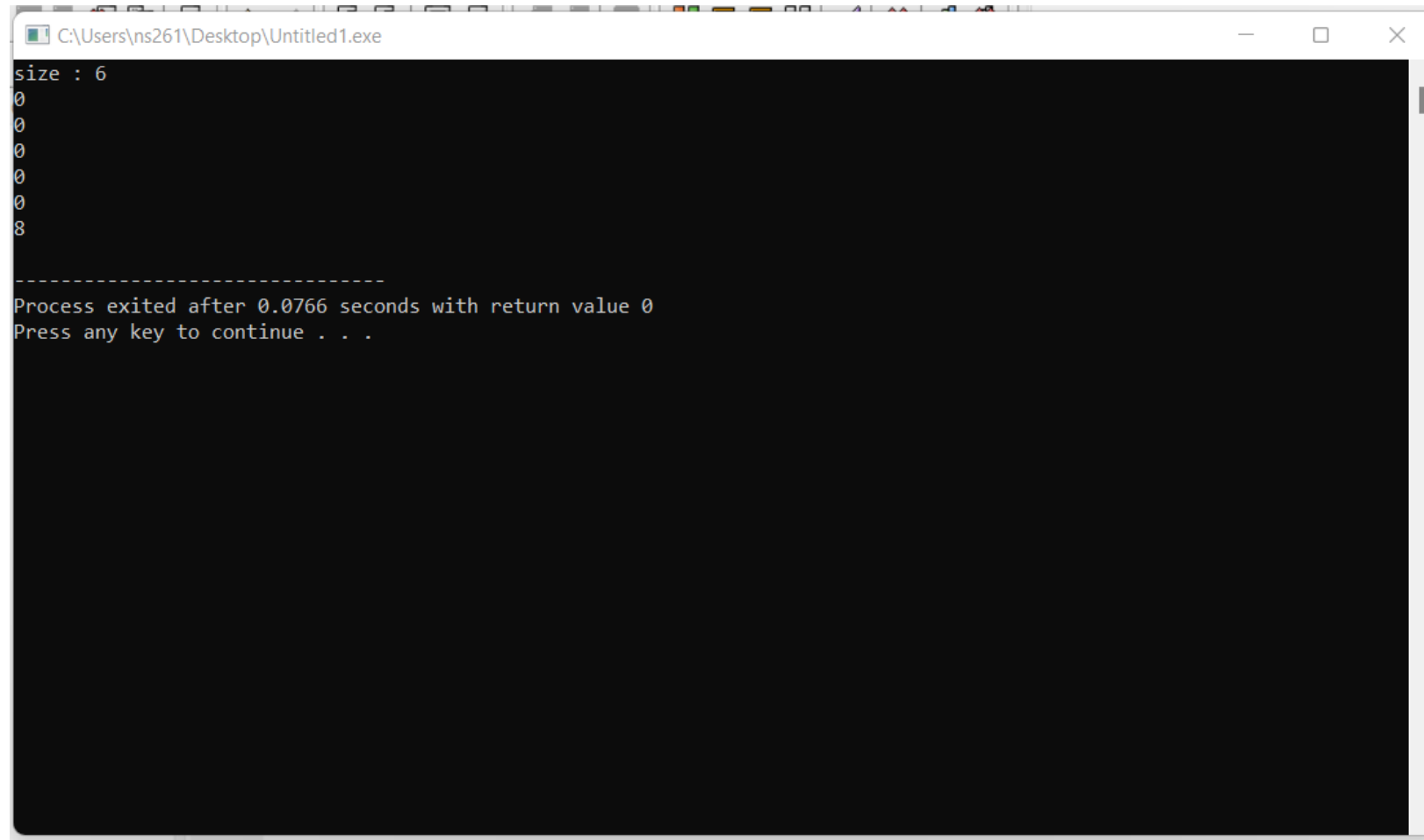
```
#include<iostream>
#include<vector>
using namespace std;

] void PrintVec(vector<int> v){
    cout<<"size : "<<v.size()<<endl;
]   for(int i=0;i<v.size();i++){
        cout<<v[i]<<endl;
-   }
- }

] int main(){
    vector<int> v(5);
    v.push_back(8);

    PrintVec(v);
- }
```

# Output



A screenshot of a Windows command prompt window titled "C:\Users\ns261\Desktop\Untitled1.exe". The window has a black background with white text. The output of the program is as follows:

```
size : 6
0
0
0
0
0
0
8

-----
Process exited after 0.0766 seconds with return value 0
Press any key to continue . . .
```

# Initializing Values in a vector of predefined size

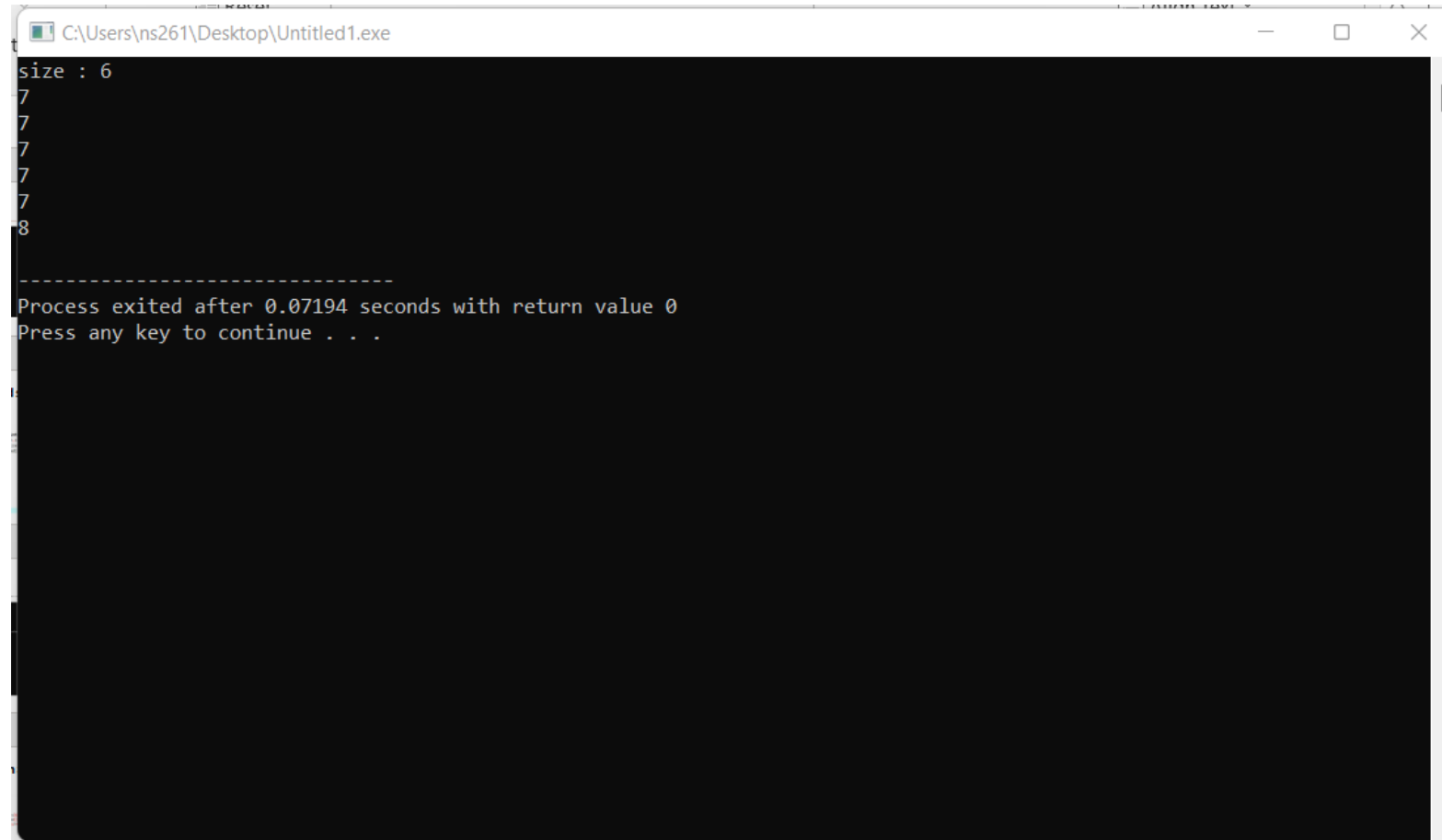
```
#include<iostream>
#include<vector>
using namespace std;

void PrintVec(vector<int> v){
    cout<<"size : "<<v.size()<<endl;
    for(int i=0;i<v.size();i++){
        cout<<v[i]<<endl;
    }
}

int main(){
    vector<int> v(5,7);
    v.push_back(8);

    PrintVec(v);
}
```

# Output



A screenshot of a Windows command prompt window titled "C:\Users\ns261\Desktop\Untitled1.exe". The window has a black background with white text. The output shows a series of numbers: "size : 6", followed by five "7"s, and then an "8". Below these numbers is a dashed line, followed by the text "Process exited after 0.07194 seconds with return value 0" and "Press any key to continue . . .".

```
C:\Users\ns261\Desktop\Untitled1.exe
size : 6
7
7
7
7
7
7
8
-----
Process exited after 0.07194 seconds with return value 0
Press any key to continue . . .
```

# pop\_back()

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

void PrintVec(vector<int> v){
    cout<<"size : "<<v.size()<<endl;
    for(int i=0;i<v.size();i++){
        cout<<v[i]<<endl;
    }
}

int main(){
    vector<int> v;
    v.push_back(8);
    v.push_back(7);

    PrintVec(v);
}
```

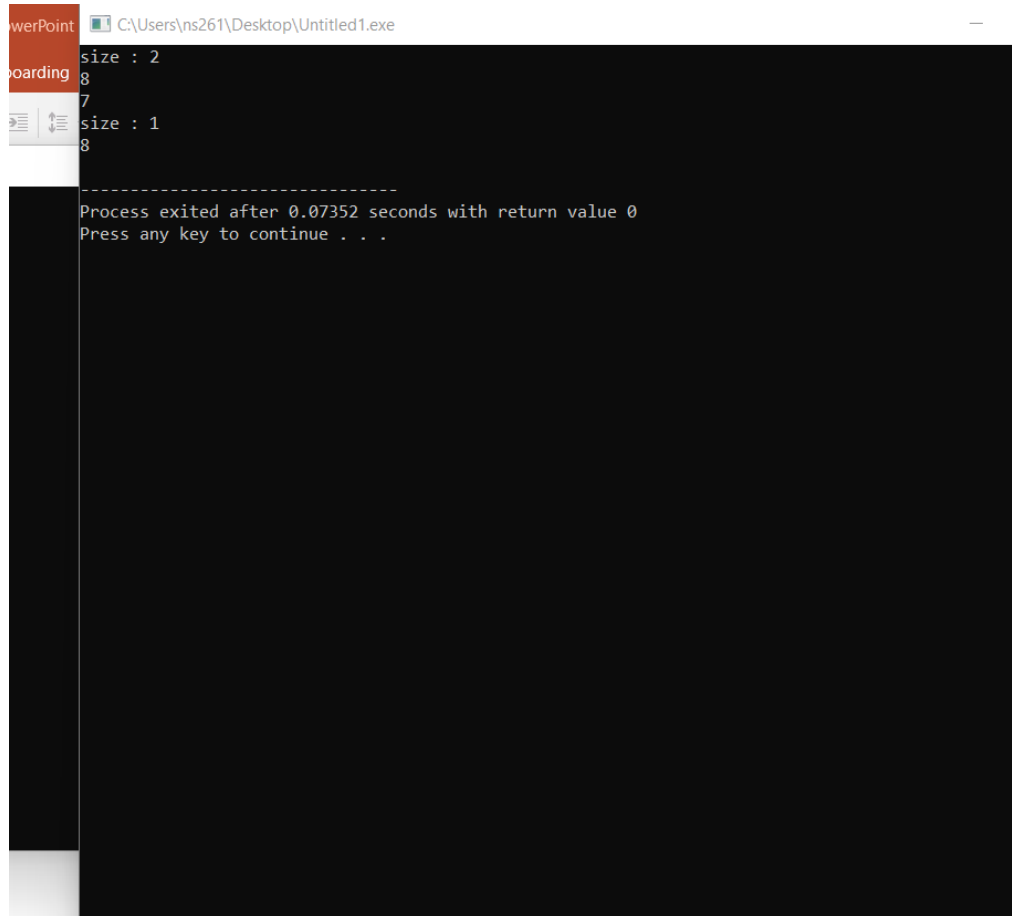
```
#include<iostream>
#include<vector>
using namespace std;

void PrintVec(vector<int> v){
    cout<<"size : "<<v.size()<<endl;
    for(int i=0;i<v.size();i++){
        cout<<v[i]<<endl;
    }
}

int main(){
    vector<int> v;
    v.push_back(8);
    v.push_back(7);

    PrintVec(v);
    v.pop_back();
    PrintVec(v);
}
```

# Output



A screenshot of a Windows command prompt window. The title bar shows 'PowerPoint' and the file path 'C:\Users\ns261\Desktop\Untitled1.exe'. The window has a standard Windows interface with a title bar, a menu bar (File, Edit, Format, View, Tools, Window, Help), and a toolbar. The command prompt itself is black with white text. The output shows several lines of text, including 'size : 2', '8', '7', 'size : 1', and '8'. A dashed line separates this from a summary line: 'Process exited after 0.07352 seconds with return value 0'. Below this, it says 'Press any key to continue . . .'. The window is partially obscured by a red 'boarding' label on the left.

```
PowerPoint C:\Users\ns261\Desktop\Untitled1.exe
boarding
size : 2
8
7
size : 1
8
-----
Process exited after 0.07352 seconds with return value 0
Press any key to continue . . .
```



# Copying a vector

```
Vector<int> v2=v;
```