```
while (ptr != NULL)
{ ptr = strtok (NULL, " ");
  cout << ptr << endl;
}
  return 0;
}
```

Vector Introduction

(container)

↳ vector is a dynamic array

↳ It is an array that can grow and shrink in size automatically depending upon the requirements

↳ It grow by doubling its size

↳ We have to include header file `#include <vector>`

```
#include <iostream>
#include <vector>
using namespace std;
int main ()
{ // Initialising the vector also creating it
  // If vector<datatype> vector_name
     vector <int> a;
```

2) vector<int> b(5,10);
// there are 5 integer in a vector (array) each having
// value 10. Fill used in initilise a vector with '0'.

3) vector<int> c(b.begin(), b.end());
// here we created a vector c having all the element of
// b starting from begining to end.

4) vector<int> d{1, 2, 3, 10, 14, 17};
// create a vector d initialise it with given values

// How to iterate over the vector (array)

1) for(int i=0; i<c.size(); i++)
{ cout<<c[i]<<",";}
cout<<"\n";

2) // using iterators.
for(vector<int>::iterator it // auto can be replaced
for(auto it = b.begin(); it != b.end(); it++)
{ cout<<(*it)<<",";}

3) // for each loop
for(auto int x : d)
{ cout<<x<<","; } cout<<"\n";

// more function on vectors

// for accessing elements from the user and add them
//the vector.

```
Vector<int> v;
int n;
cin>>n;
for(int i=0; i<n; i++)
{ int num;
  cin >> num;
  v.push_back(num);
}
```

// v.push_back add a no. at the end of the vector.

```
for(auto x:v)
{ cout<<x<<",";
}
```

// size of vector (How many element the vector have)

```
cout<< v.size()<<"\n";
```

// size of underlined array.

```
cout<<v.capacity()<<"\n";
```

// How much the vector can expand in worst case
// according to available memory in system

```
cout<<v.max_size()<<"\n";
```

Vector 02 - Methods

```
#include <iostream>
#include <vector>
using namespace std;
int main()
{
    // create and initialize a vector
    vector<int> d{1, 2, 3, 10, 14};
    // adding an element at the end. (T = O(1))
    d.push_back(16);
    // for accessing / declaring printing all the vector elements
    for (int x : d)
    { cout << x << ", "; }
// O/P → 1, 2, 3, 10, 14, 16,
    // for deleting the last element from vector
    d.pop_back();
    // O/P → 1, 2, 3, 10, 14,   T = O(1):
    // Inserting some element in the middle
// vectorname.insert(vectorname.begin() + Pos, element)
    d.insert(d.begin() + 3, 100);
// It will insert the element at 3rd pos starting from 0.
// O/P→ 1, 2, 3, 100, 10, 14,
// also, we can add more than 1 elements.
// v.insert(d.begin() + 3, no.of elements, value)
```

d.insert(d.begin()+3, 4, 100);

// O/P → 1, 2, 3, 100, 100, 100, 100, 10, 14

// It will insert 4 elements after pos 3 each having value 100

// Time comp → O(n)

// Erase some of the elements from the middle.

d.erase(d.begin()+3);

// O/P → 1, 2, 3, 100, 100, 100, 10, 14,

// We can erase a range of elements

d.erase(d.begin()+2, d.begin()+5);

// O/P → 1, 2, 100, 10, 14 This will erase 3 elements

// size

cout << d.size() << "\n";

// capacity

cout << d.capacity() << "\n";

// resize operation (to resize the vector)

d.resize(8);

cout << d.capacity() << "\n";

// 1, 2, 100, 10, 14, 0, 0, 0 → make the size of the vectors

// remove the vector

d.clear();

```
// getting the 1st element of the vector.
cout << d.front();
// getting the last element of the vector
cout <<d.back();
// To avoid doubling, we will use reserve functh
    v.reserve(1000);
// its like predifing the sig of array.
return 0;
}
```