Tishk International University
IT Department
Course Code: IT-117

Programming I

Lecture 7

Arrays

Fall 2023

Hemin Ibrahim
hemin.ibrahim@tiu.edu.iq

Outline

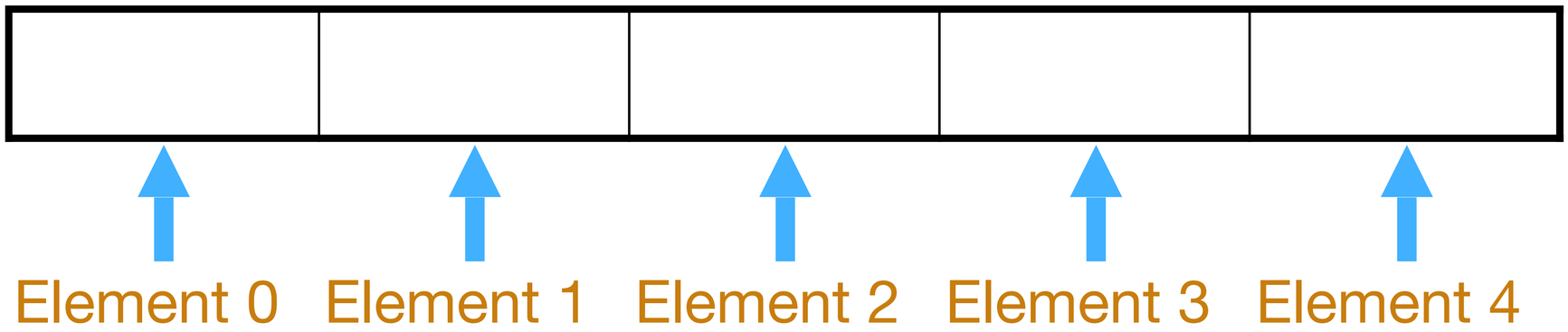
- Concept of Array
- Defining Array
- Array Initialization
- Inputting Array Contents
- Processing Array Contents
- Displaying Array Contents

Objectives

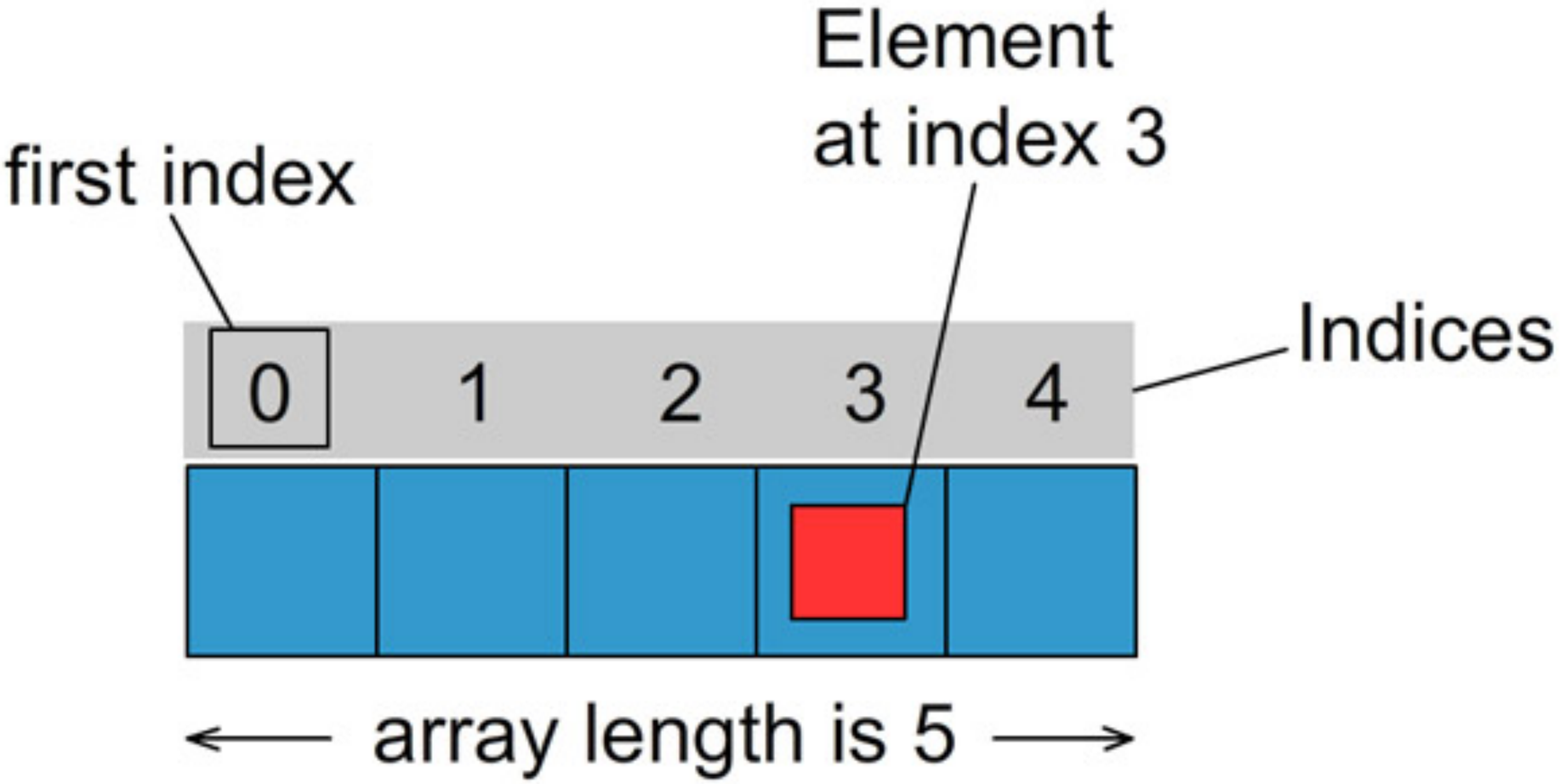
- Define what an array is and how it differs from individual variables.
- Demonstrate how to declare and define arrays in C++ using appropriate syntax.
- Describe different methods of initializing arrays in C++, including static and dynamic initialization.
- Demonstrate methods for displaying array contents to the console using loops.

Arrays

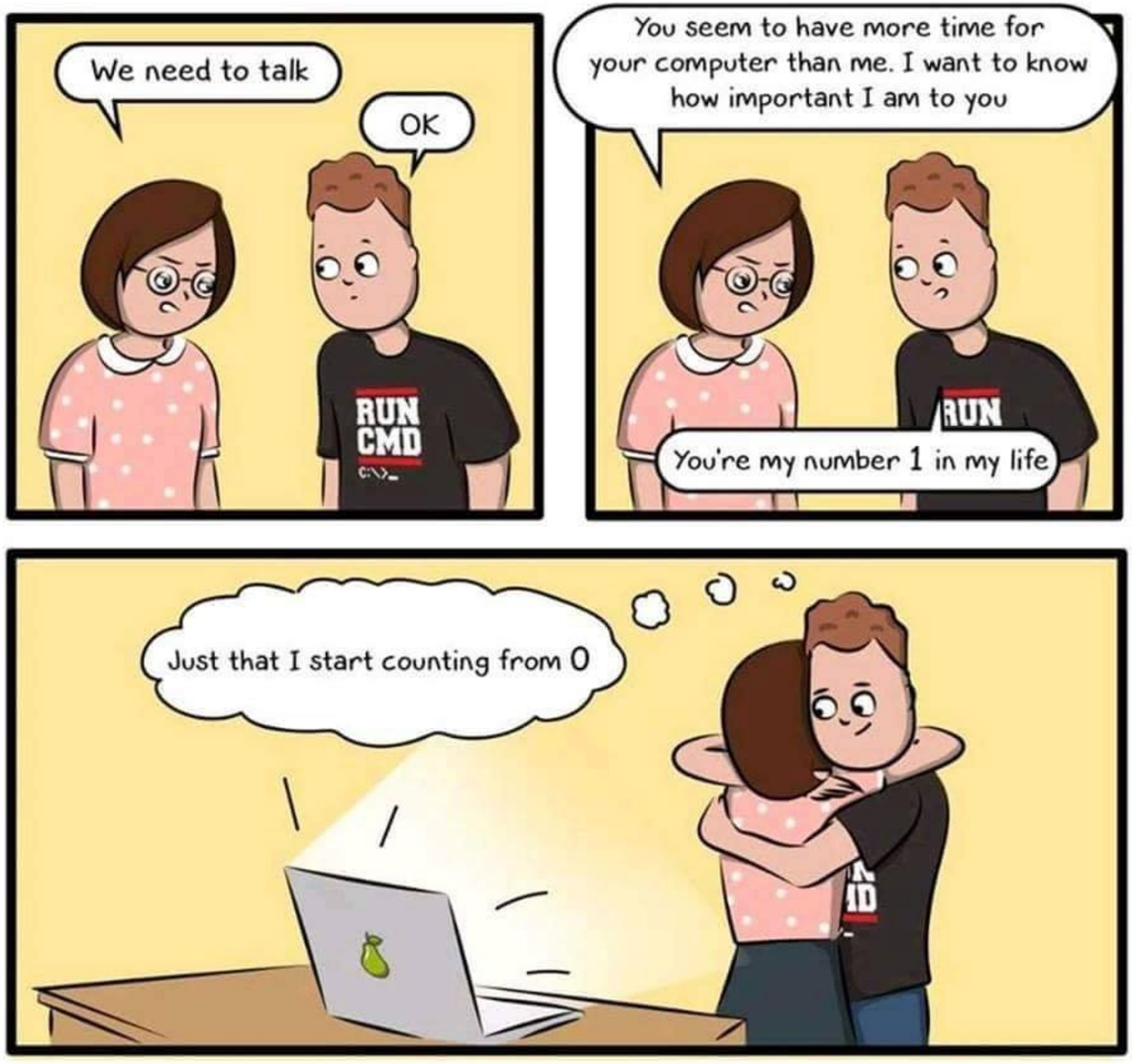
- Arrays are variables capable of storing multiple values of the same type.
- Values within arrays are stored in adjacent memory locations.
- Array declaration involves using the [] operator.
- Example: int A[5];

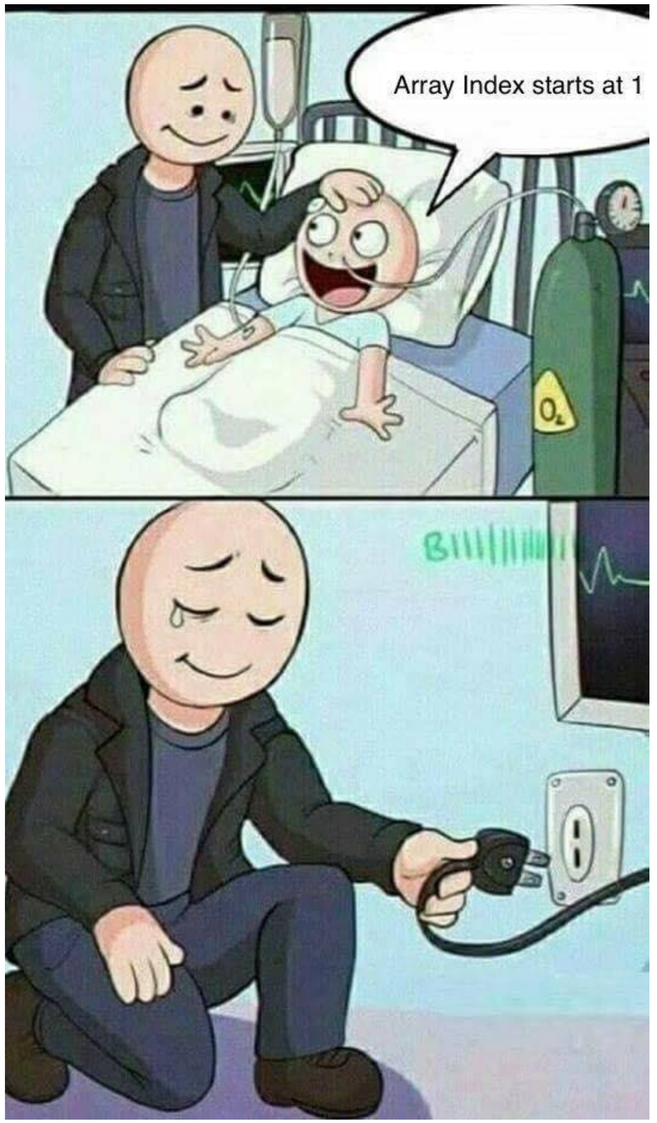


Arrays


Element
at index 3
first index
Indices
0
1
2
3
4
array length is 5

Arrays


We need to talk
OK
RUN CMD
You seem to have more time for your computer than me. I want to know how important I am to you
RUN
You're my number 1 in my life
Just that I start counting from 0
fb.com/programmingjokes
NERD 4 LIFE .studio


Array Index starts at 1

Declaring Arrays

```
datatype arrayName[arraySize];
```

```
int numbers[5]; //holds 5 integer cells
```

- int is the data type of the array elements
- A is the name of the array
- 5 is the number of elements. It shows the number of elements in the array.

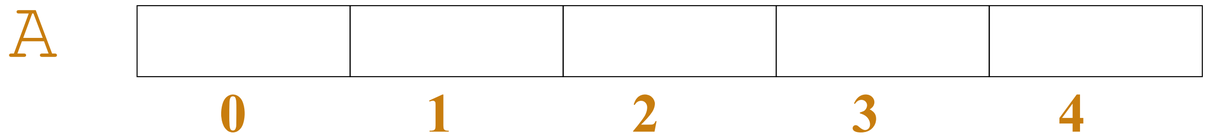
```
double volumes[10];//holds 10 double cells
```

Array Size Declarators

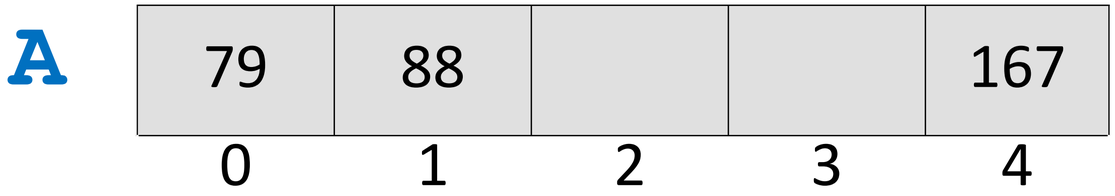
Array declaration	Number of elements	Size of each element	Size of the array
`char letter[26];`	26	1 byte	26 bytes
`short ring[100];`	100	2 bytes	200 bytes
`int mile[84];`	84	4 bytes	336 bytes
`float temp[12];`	12	4 bytes	48 bytes
`double distance[1000];`	1000	8 bytes	8,000 bytes

Accessing Array Elements

Array elements, accessed using the array name and index, can be treated like regular variables.




```
5       int A[5];
6       A[0] = 79;
7       cout << A[0];
8       cin  >> A[1]; //user puts 88
9       A[4] = A[0] + A[1];
```



Initializing Arrays

Arrays in C++ can be initialized in several ways:

- Initialization with Assignment Statements: Individual array elements can be initialized during program execution using assignment statements.

```
A[0] = 79;
A[1] = 82;
```

- Initialization at Array Definition with an Initialization List:

```
int A[5] = {79,82,91,77,84};
```

- Special Cases for Initialization Lists:

```
int A[5] = {0};//all have zero value
int A[5] = {4};//only the first element has value 4, others
are zero
```

Implicit Array Sizing

Can determine array size by the size of the initialization list

```
int quizzes[]={12, 17, 15, 11};
```

Must use either array size declarator or initialization list when array is defined

Inputting Array Contents

cin can be used to input values from keyboard and store these values into an array element.

```
int A[5]; // Define 5-cells array
cout << "Enter first number ";
cin >> A[0];
```

Processing Array Contents

- Array elements can be treated just like regular variables of the same type as the array.
- You can perform arithmetic operations using array elements.
- Array elements can be used in relational expressions, like compare elements.
- They can also be used in logical expressions, like AND, OR, and NOT operations.

```
int numbers[] = {5, 2, 3, 4, 5};
int sum = numbers[0] + numbers[1];
int product = numbers[2] * numbers[3];
if (numbers[0] < 0) {
    cout<<numbers[0] * -10;
} else {
     cout<<numbers[0] * 7;
}
```

Displaying Array Contents and the size of array

cout can be used to display value of the value of an array element

```
#include <iostream>
using namespace std;
int main() {

    string A[]={"Hawler", "Slemani","Duhok", "Halabja"};
    cout<<"A[1]= " << A[1] <<endl;

    return 0;
}
```

```
int size = sizeof(A)/sizeof(A[0]);
cout<<size<<endl; // 4
```

Array Subscripts

- Array subscript (index) can be an **integer constant**, **integer variable**, or **integer expression**

```cpp
#include <iostream>
using namespace std;
int main() {
  // Declare an array
  int arr[] = {10, 12, 0, 4, 3};

  // Accessing array elements using integer constants
  cout << "Element at index 0: " << arr[0] << endl; // Output: 10
  cout << "Element at index 2: " << arr[2] << endl; // Output: 0

  // Accessing array elements using integer variables
  int index = 3;
  cout << "Element at index " << index << ": " << arr[index] << endl; // Output: 4

  // Accessing array elements using integer expressions
  int i = 1, j = 2;
  cout << "Element at index " << i+j << ": " << arr[i+j] << endl; // Output: 4

  return 0;
}
```

-


How do you execute a code block 100 times?
I just copy-paste it 100 times.
imgflip.com

Inputting All Array Elements

To access each element of an array

- Use a loop
- Let the loop control variable be the array subscript
- A different array element will be referenced each time through each cycle of the loop

```
int A[4];
int size = sizeof(A)/sizeof(A[0]);

for(int i=0;i<size;i++){
    cout<<"Enter number to array ";
    cin>>A[i];
}
```

Displaying All Array Elements

To display each element of an array, use a loop

```
#include <iostream>
using namespace std;
int main() {

   float A[5]={ 5.76 , 8.1 , 0.5 , 6.1 , 3.5 };

   int size = sizeof(A)/sizeof(A[0]);

    for(int i=0;i<size;i++){
       cout<<A[i]<<endl;
   }
    return 0;
}
```

Sample input and output program of array

```cpp
#include <iostream>
using namespace std;
int main() {

   int A[4];
   int size = sizeof(A)/sizeof(A[0]);
   for(int i=0;i<size;i++){
       cout<<"Enter number to array: ";
       cin>>A[i];
   }

    for(int i=0;i<size;i++){
       cout<<A[i]<<endl;
   }
    return 0;
}
```

Output

```
Enter number to array: 4
Enter number to array: 3
Enter number to array: 2
Enter number to array: 0
4
3
2
0
```

Strings and string Objects

String is a special type of array of characters.

It Can be processed using array name

- Entire string at once, or
- One element at a time by using a subscript

```
string city= “Hawler";
```

Strings and string Objects

```
#include <iostream>
using namespace std;
int main() {
    string city="Hawler";

    for(int i=0;i<city.length();i++){
       cout<<city[i]<<endl;
    }
    return 0;
}
```

Output

```
H
a
w
l
e
r
```

Example

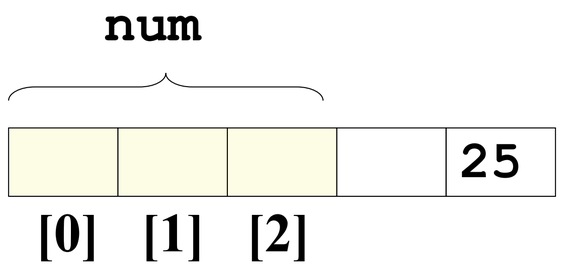
```
// This program displays the number of days in each month. It uses an
 // array of string objects to hold the month names and an int array
 // to hold the number of days in each month. Both are initialized with
 // initialization lists at the time they are created.
#include <iostream>
using namespace std;
#include <iomanip>
int main(){
     const int NUM_MONTHS = 12;
     string name[NUM_MONTHS] = {"January", "February", "March","April", "May",
         "June","July", "August", "September","October", "November", "December"};

    int days[NUM_MONTHS] = { 31, 28, 31, 30,31, 30, 31, 31,30, 31, 30, 31 };

    for (int month = 0; month < NUM_MONTHS; month++){
        cout << setw(9) << left << name[month] << " has ";
        cout << days[month] << " days.\n";
    }

    return 0;
}
```

```
January   has 31 days.
February  has 28 days.
March     has 31 days.
April     has 30 days.
May       has 31 days.
June      has 30 days.
July      has 31 days.
August    has 31 days.
September has 30 days.
October   has 31 days.
November  has 30 days.
December  has 31 days.
```

NOTE 1: No Bounds Checking

- There are no checks in C++ that an array subscript is in range
- An invalid array subscript can cause program to overwrite other memory locations

```
int num[3]; //composed of =>num[0], num[1], num[2]
cout<<sizeof(num)/sizeof(num[0])<<endl; // 3
int i = 4;
num[i] = 25;//we don't have num[4]
cout<<sizeof(num)/sizeof(num[0])<<endl; //3
```



NOTE 2: Using Increment and Decrement

When using ++ and -- operators, don't confuse the element with the subscript

```
A[i]++; // adds 1 to A[i]
A[i++]; // increments i, but has no effect on A
```

```
int score[5] = {7, 18, 9, 21, 11};
++score[2]; // Pre-increment operation on the value in score[2]
score[4]++; // Post-increment operation on the value in score[4]
```

NOTE 3: Copying One Array to Another

```
int A1[]= {1,2,3,4,5};
int A2[5];
```

- Cannot copy with an assignment statement:

```
A2=A1 ;  //Not allowed
```

- But we can copy with an assignment statement inside a loop:

```
#include <iostream>
using namespace std;
int main() {

    int A1[]= {1,2,3,4,5};
    int A2[5];
    int sizeA1 = sizeof(A1)/sizeof(A1[0]);

    for (int i=0; i < sizeA1; i++) {
        A2[i] = A1[i];
    }

    return 0;
}
```

Search inside an array

Write a program to search for an input inserted by user

```
#include <iostream>
using namespace std;
int main() {
    int myArray[]={3,4,7,1,9};
    int size = sizeof(myArray)/sizeof(myArray[0]);
    int x;
    bool found;
    cout<<"Search for a number: ";
    cin>>x;
    for(int i=0;i<size;i++){
        if(myArray[i]==x){
            found=true;
            break;
        }
    }

    if(found){
        cout<<"Number found.";
    } else {
        cout<<"Number is not exist.";
    }
    return 0;
}
```

Vowel counter application

```cpp
#include <iostream>
#include <string>
using namespace std;
int main() {
    string myText;
    int counter=0;
    char ch;
    cout<<"Input any sentences to count number of Vowels:"<<endl;
    getline(cin,myText);
    for(int i=0;i<myText.length();i++){
        ch=tolower(myText[i]);
        if(ch=='a' || ch=='e' || ch=='i' || ch=='o' || ch=='u'){
            counter++;
        }
    }
    cout<<"There are "<<counter<<" vowels in the Text.";
    return 0;
}
```

Thank You