ARRAYS

HOW TO STORE THE MARKS OF 10 STUDENTS OR 100 STUDENTS??

10 variables for
10 students
S1,S2,S3.....S10

What about
100
or 1000
students??

Introduction To Array

- **An array data structure is a collection of similar elements stored in CONTIGUOUS memory locations to compute under a single variable name.**
- **e.g. Array of characters , Array of Integers, etc**

{15, 12, 20, 14}

{'A', 'C', 'M'}

Which is not an array??

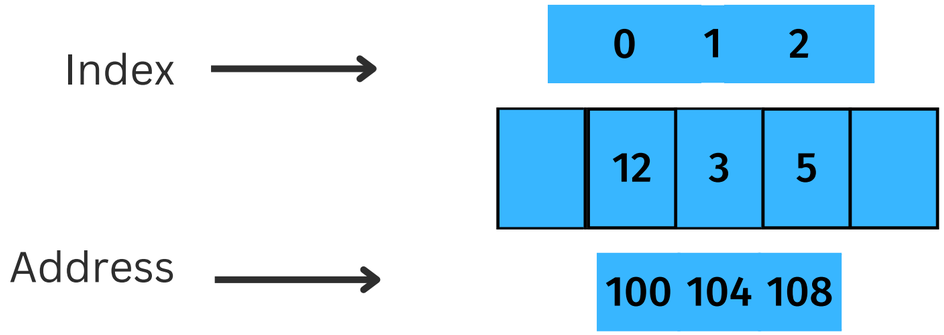
{'G', '2', '$'}

{'A', '20', 30}

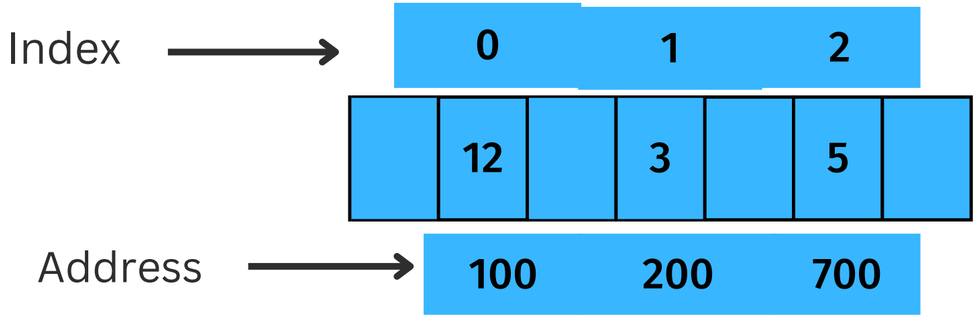
CONTIGUOUS

A single contiguous section of memory blocks is allotted depending on its requirements.

int arr[3]={12, 3, 5};



CONTIGUOUS



RANDOM

SYNTAX OF ARRAY IN CPP

In C++, we can declare an array by simply specifying the data type first and then the name of an array with its size.

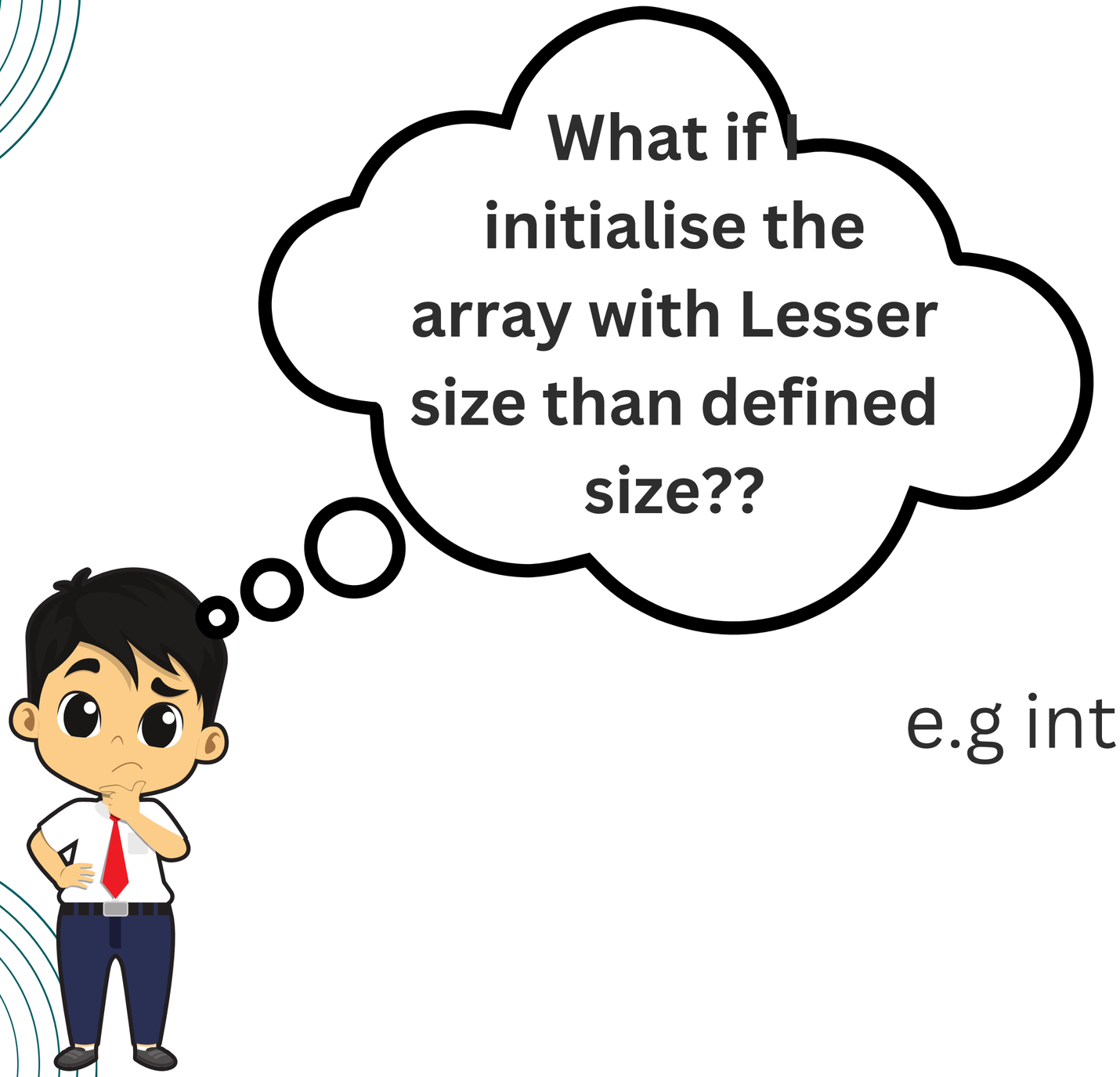
```
data_type array_name[Size_of_array];
```

Example

```
int arr[5];
```

INITIALIZATION OF ARRAY

1. Initialize an array with values.
2. Initialize Array with values & without size.
3. Initialize array after declaration(using loop)
4. Initialize an array partially.
5. Initialize the array with zero



e.g int arr[5]={20, 14, 1};

How to calculate total memory allocated by array?

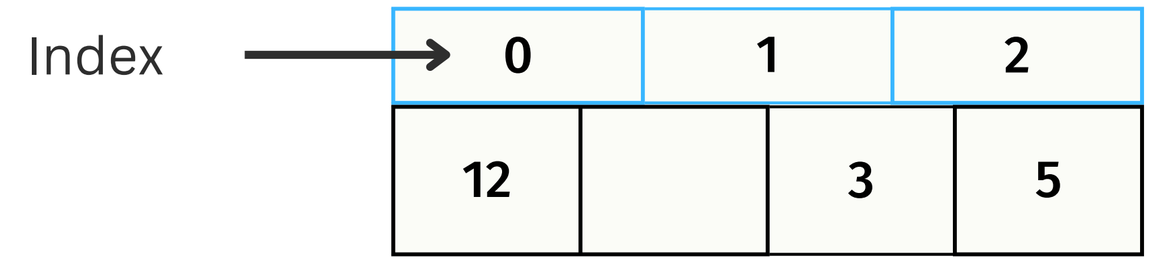
Memory required= (size_of_array)*(size_of_data_type)

ACCESING ARRAY ELEMENTS

- **Method 1**
  - To access the elements of array , we can use the Index of array.
  - Array index starts from 0.

K= arr[index]

int arr[3]={12, 3, 5};

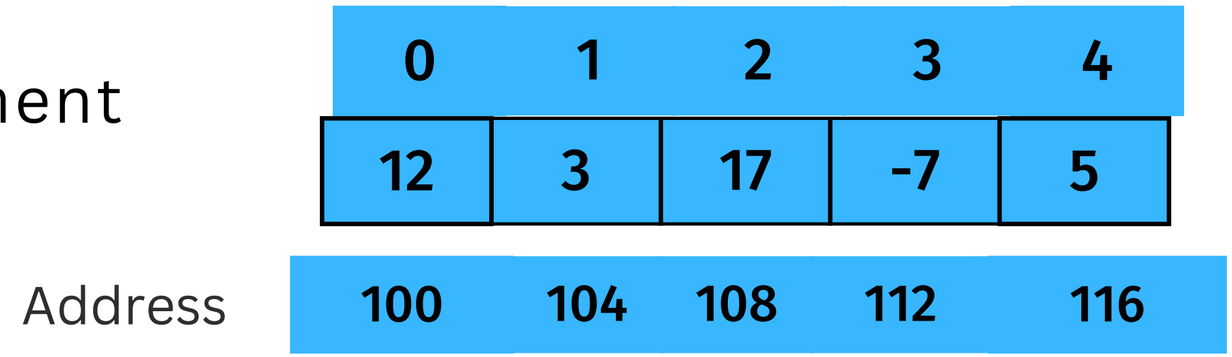


Why index starts from 0?

ACCESING ARRAY ELEMENTS

- **Method 2**
  - To access the loaction/address of element (arrName+index)
  - To access the value *(arrName+index)

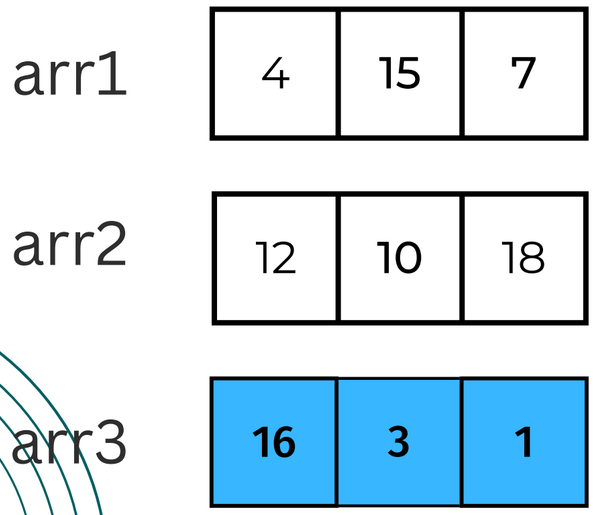
int arr[ ]={12, 3, 17, -7,5};

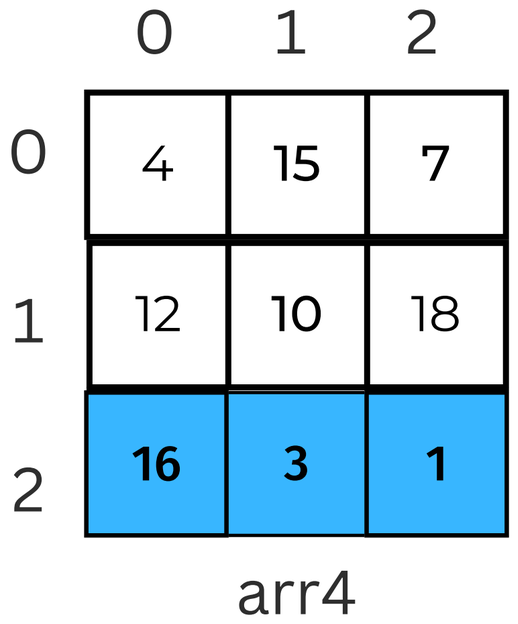


arr

2D ARRAY
OR
MULTIDIMENSIONAL ARRAY

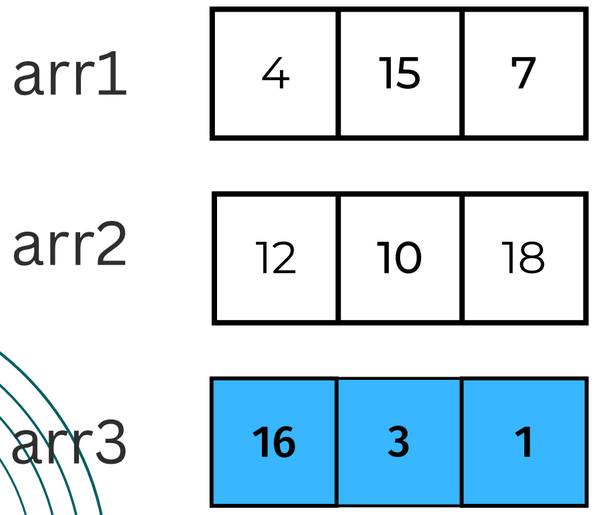
- An array of an array, known as a multidimensional array.

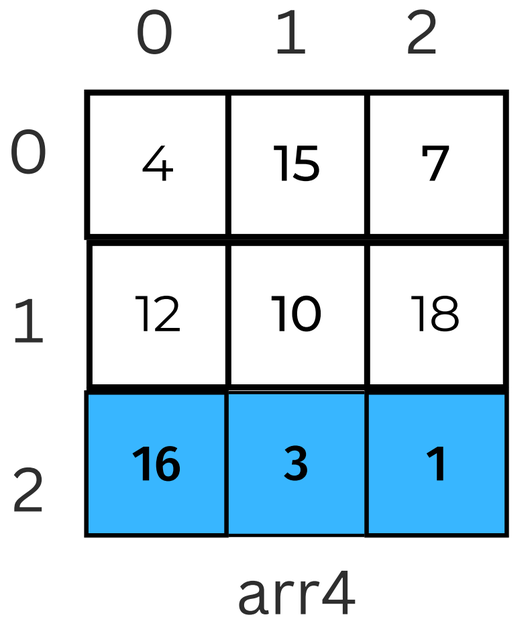




2D ARRAY
OR
MULTIDIMENSIONAL ARRAY

- An array of an array, known as a multidimensional array.

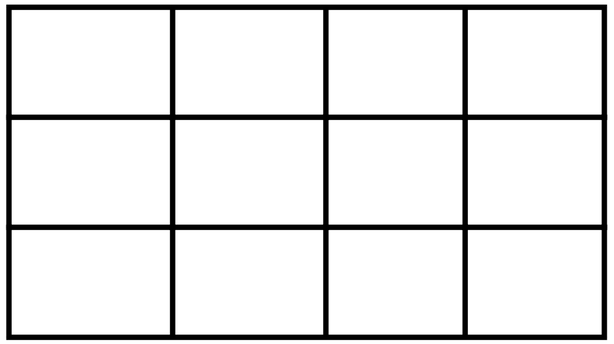




SYNTAX OF 2D ARRAY IN CPP

1. data_type array_name[row][coloumn];

int arr[3][4];



SYNTAX OF 2D ARRAY IN CPP

2. data_type array_name[row][coloumn]={<initialisation>];

// initializes an 2D array with 12 values given from the user

1) int arr[3][4] = {67, 40, 58, 22, 53, 62, 21, 55, 87, 78, 15, 61};

2) int arr[3][4] = { {67, 40, 58, 22}, // row 0
{53, 62, 21, 55}, // row 1
{87, 78, 15, 61} // row 2
};

67	40	58	22
53	62	21	55
87	78	15	61

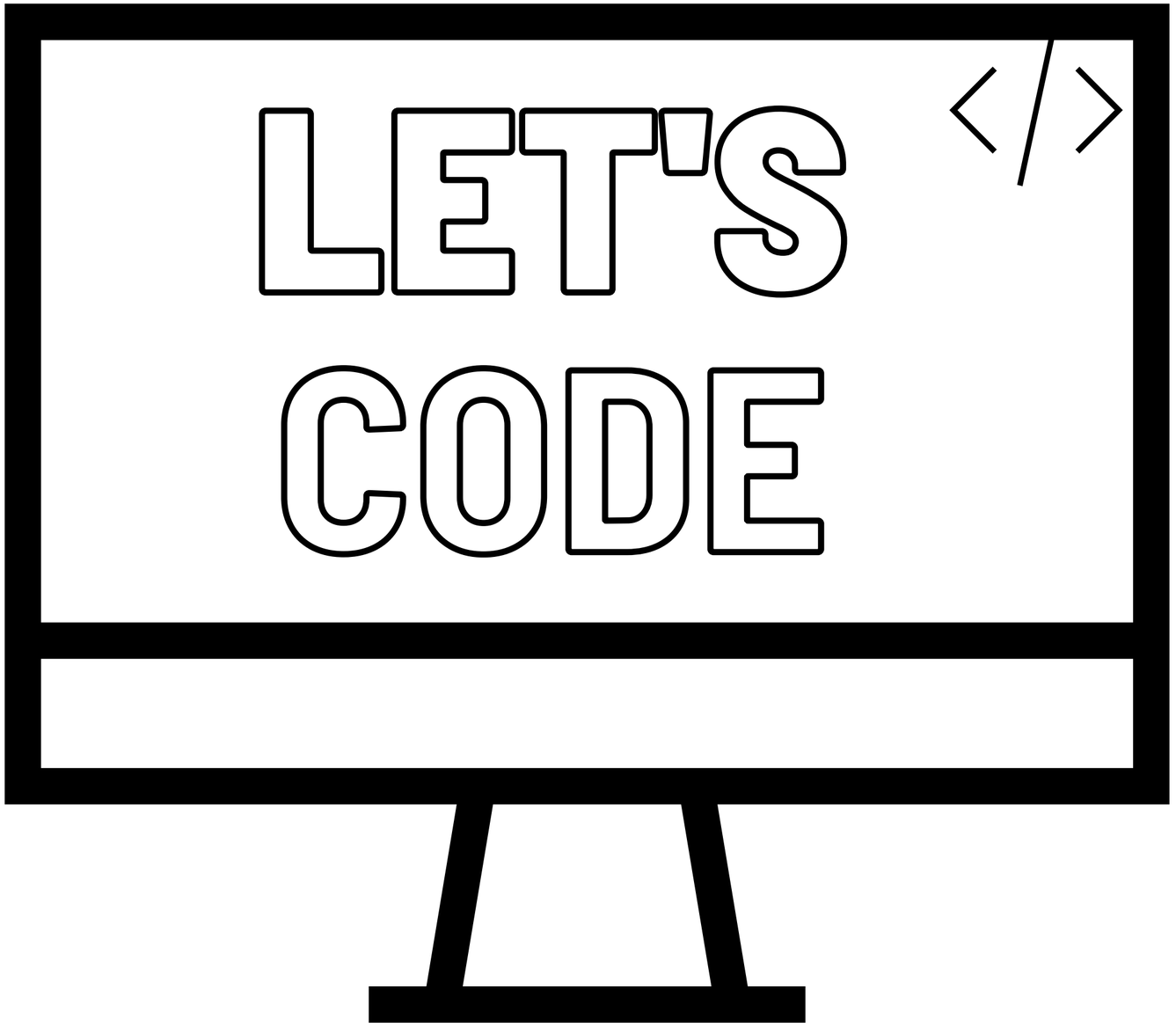
ADVANTAGES AND DISADVANTAGES

ADVANTAGES

- Traversing an array is a very simple process.
- Direct access is possible.
- Searching process is easy. e.g. binary search
- 2D Array is used to represent matrices.

DISADVANTAGES

- Array size is fixed
- Insertion and deletion are not easy in Array
- There will be chances of wastage of memory


LET'S
CODE

GAME OF PREFIX AND SUFFIX

Given an Array of N elements . Let P(i) represent the Prefix till ith index and S(i) represent the suffix from ith index. You have to find the first index at which sum of Prefix and Suffix is maximum.

$1<=N<=10^5$

Input Constraint :

N -> Number of Elements in an array

Next N values will be the Array elements

Output Constraint :

Print Index that giving an maximum sum

Note : Assuming 1 based Indexing

MAXIMUM SUBARRAY

Given an integer array nums, find the subarray

with the largest sum, and return its sum.

Example 1:
Input: nums = [-2,1,-3,4,-1,2,1,-5,4]
Output: 6
Explanation: The subarray [4,-1,2,1] has the largest sum 6.

Brute Force

```cpp
int maxSubarraySum(int arr[], int n)
{    int maxi = INT_MIN;

     for (int i = 0; i < n; i++) {
          int sum = 0;
          for (int j = i; j < n; j++) {
               sum += arr[j];

               maxi = max(maxi, sum);
          }
     }

     return maxi;
}
```

Optimal approach

KADANE'S ALGORITHM

- Kadane's algorithm is a dynamic programming approach used to solve the maximum subarray problem, which involves finding the contiguous subarray with the maximum sum in an array of numbers.

- The algorithm was proposed by Jay Kadane in 1984 and has a time complexity of O(n).

STRINGS

Introduction to string

string is sequence of characters stored in contiguous memory locations, terminated by a special character called the null character ‘\0’.

Null character is used to represent the end of the string or end of an array.

1. strings are used to store words or text.
2. they are also used to store data or other information

CHARACTER ARRAY

1.Need to know size

2.Larger size required for operations

3.No terminating extra character

4.size of array has to be allocated statically

STRINGS

1. No need to know size

2.Performing operations like concatenation & append is easier.

3.Terminated with extra character ‘\0’

4.memory is allocated dynamically.

ways to define string

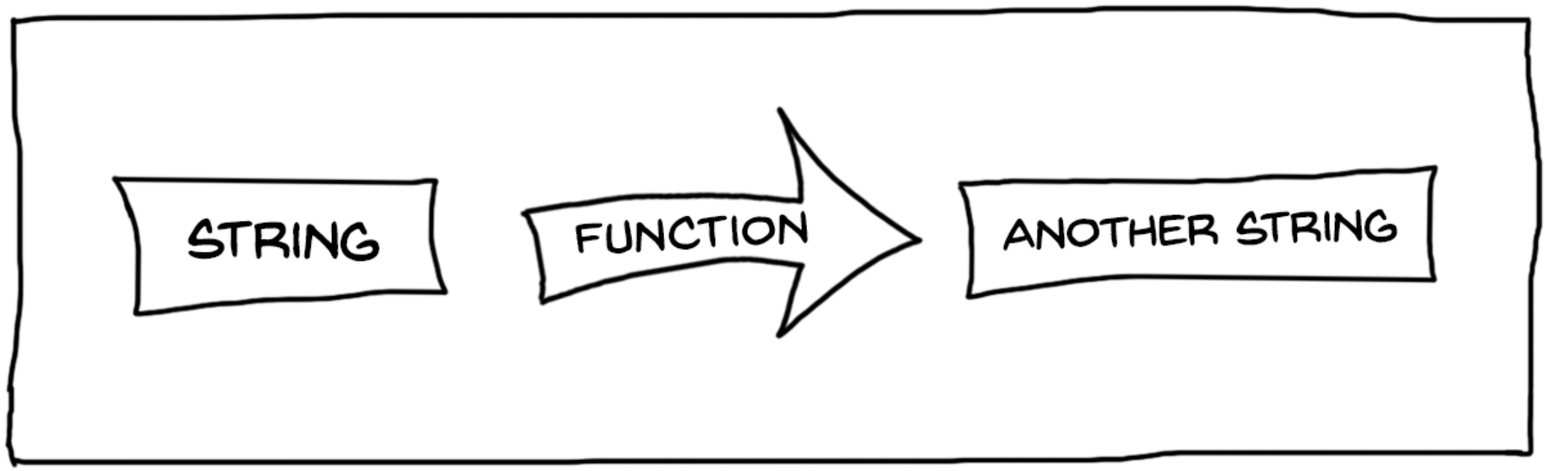
```cpp
#include <iostream>
using namespace std;

int main()
{
    char s1[] = "DSA LAUNCHPAD";
    cout << s1 << endl;
    string str1("ACM\n");
    cout << str1<<endl;
}
```

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

int main()
{
    string str2(5, 'P');
    cout << str2<<endl;
    string s2;
    cout<<"Enter String\n"<<endl;
    cin>>s2;
    cout<<"String is: "<<s2<<endl;
}
```

OPERATIONS ON STRING



INPUT FUNCTIONS

Function	Definition
getline()	Used to store a stream of characters
push_back()	Used to input a string at the end of the string
pop_back()	Used the delete the last character from the string

WHY DO WE NEED INPUT FUNCTIONS?

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

int main()
{

   string str1;
   char ch1;
   getline(cin,str1);
   cout<<"string:"<<str1<<endl;
   cout<<"\nenter a character to append at end";
   cin>>ch1;
   str1.push_back(ch1);
   cout<<"After performing operation :"<<str1<<endl;
   str1.pop_back();
   cout<<"After performing operation :"<<str1<<endl;
}
```

CAPACITY FUNTIONS

Function	Definition
capacity()	Returns the capacity of string equal to or more than size of string. Additional space is allocated for further modifications
resize()	Used to resize the size of string.
length()	Finds the length of string
shrink_to_fit()	Decreases the capacity of string and makes it equal to the minimum capacity of the string. Saves memory

```cpp
#include<iostream>
#include<string>
using namespace std;
int main()
{
    string str1="Good Evening";
    cout<<"String1: "<<str1<<endl;
    cout<<"Capacity of string: "<<str1.capacity()<<endl;
    cout<<"Length of string: "<<str1.length()<<endl;
    str1.shrink_to_fit();
    cout<<"Capactity of str1 after shrink_to_fit: "<<str1.capacity()<<endl;
    str1.resize(4);
    cout<<"Resized string: "<<str1;
    return 0;
}
```

ITERATOR FUNCTIONS

Function	Definition
begin()	Returns an iterator to the beginning of the string
end()	Returns an iterator to the next to the end of string
rbegin()	Returns a reverse iterator pointing at the end of string
rend()	Returns a reverse iterator pointing to the previous beginning of string
cbegin() cend()	returns constant iterator pointers at beginning and end respectively. Cannot be used to modify the contents it is pointing to.
crbegin() crend()	Constant reverse iterators

```cpp
#include<iostream>
#include<string>
using namespace std;

int main()
{
    string s1="DSA Launchpad";
    for(auto it=s1.begin();it!=s1.end();it++)
        cout<<*it;
    cout<<endl;
    for(auto rit=s1.rbegin();rit!=s1.rend();rit++)
        cout<<*rit;
}
```

MANIPULATING FUNTIONS

Function	Definition
copy(”char array”,len,pos)	Copies substring to the target char array. It takes three parameters char array, length to be copied, start position in string to start copying
swap()	Swaps one string with another

```cpp
#include<iostream>
#include<string>
using namespace std;

int main()
{
    string str1="Have a good day";
    char str2[40];
    str1.copy(str2,4,7);
    cout<<"New copied character array: "<<str2<<endl;
    string s1,s2;
    s1="Nohara";
    s2="Shinchan";
    cout<<"Before Swapping \nFirst String: "<<s1<<"\nSecond String: "<<s2<<endl;
    swap(s1,s2);
    cout<<"After Swapping \nFirst String: "<<s1<<"\nSecond String: "<<s2<<endl;
    return 0;
}
```

OTHER FUNCTIONS

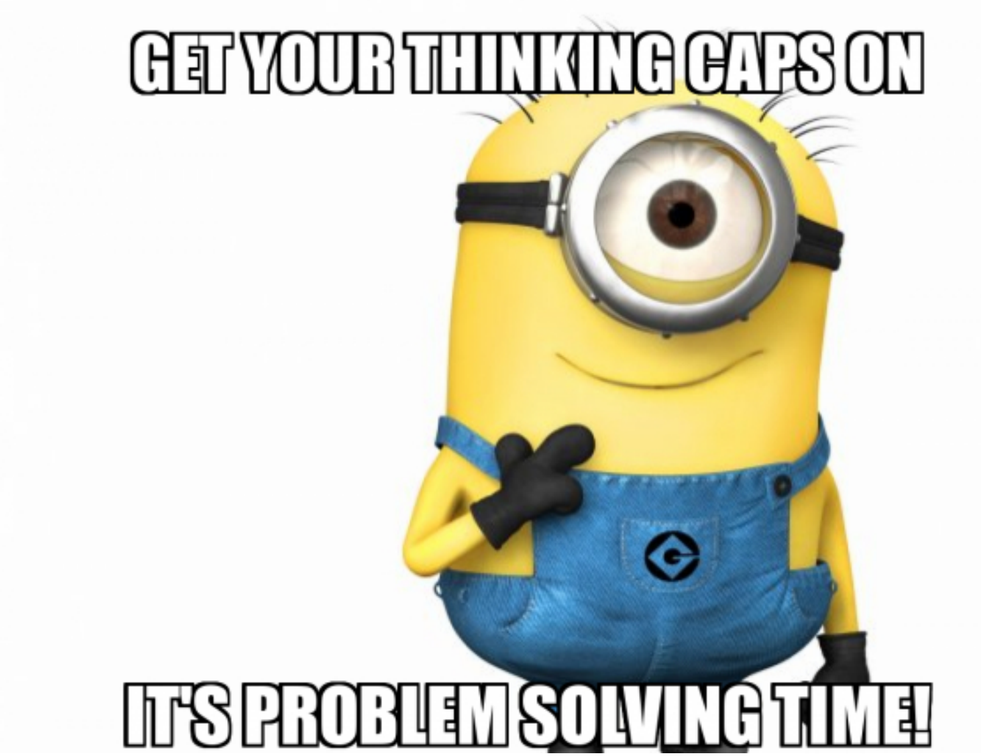
Function	Definition
strcmp(str1,str2)	Used to compare two strings. Returns 0,greater than 0,positive or negative value
substr(start,len)	Used to obtain a substring. Takes position that is start of new substring and length of new substring

- append(),
- + for concatenation
- To include special characters in string we make use (\) backslash character e.g.\' \" \\

```cpp
#include<iostream>
#include<cstring>
using namespace std;

int main()
{
    char str1[]="Hello";
    char str2[]="Hello";
    int i=strcmp(str1,str2);
    cout<<i<<endl;
    string str3="Hi Doraemon",res;
    res=str3.substr(3,8);
    cout<<res;
    return 0;
}
```

Problem Solving Time...



Roman To Integer

Given a string in roman numerals format (s) your task is to convert it to an integer . Various symbols and their values are given below.

I= 1

V= 5

X= 10

L= 50

C= 100

D= 500

M= 1000

Check if two strings are k-anagrams or not

Given two strings of lowercase alphabets and a value k, the task is to find if two strings are K-anagrams of each other or not.

THANK YOU!