char Arrays

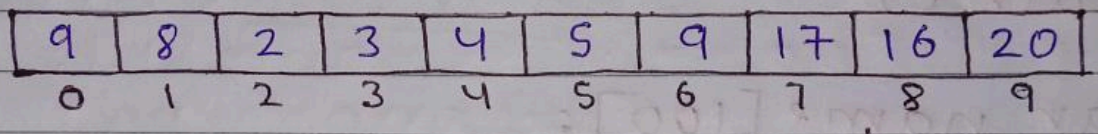
25-02-2023

Date.....

```
int arr[10];
```

↳ this statement creates a array which contain 10 block in the memory & these blocks are of integer types.

every data is of int type.

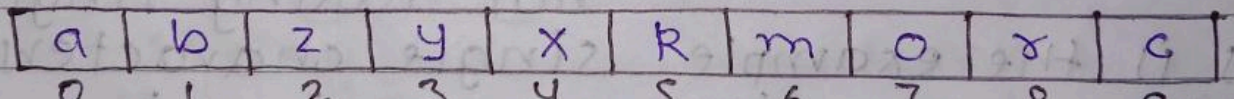


↓ each block takes 4 bytes.

```
char arr[10];
```

↳ using this statement, we can create an array which contain 10 blocks in the memory & these blocks are of character type.

every data is of char type



↓ we can store 'a' to 'z', 'A' to 'Z', '0' to '9', '+', '-', '*', any type of character in this block.

↓ each block takes 1 byte.

we can take input to char array with 2 methods :-

single character input

whole array input

```
int main()
{
    char name[100];
    cout << "enter name : ";
    cin >> name;
    cout << "Name is: " << name ;
    return 0;
}
```

this is the example of whole array input, because we are not taking single single characters as an input.

this is the example of single character input

```
int main()
{
    char ch[100];
    ch[0] = 'a';
    ch[1] = 'b';
    cin >> ch[2];
    cout << ch[0] << ch[1] << ch[2];
    return 0;
}
```

Teacher's Sign .....

Date.....

when we take input of whole array like this :-

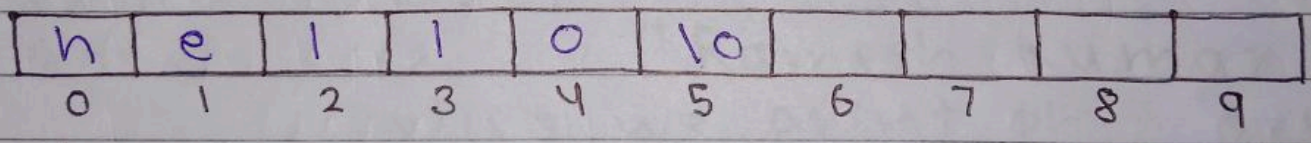
```
\0      cin >> name;
```

then by default a null character ('\0') store at the end of the input.

For example :-

```
char name[10];
cin >> name;
```

→ suppose user gives "hello" as an input. Then it will store in array like this :-



string that user gives

this null character stores by default in the end of the input string.

* Null character shows that the string is terminated.

Date.....

* Printing the null value:-

```
int main()
{
    char name[10];
    cout << "\nEnter name : ";
    cin >> name;
    for(int i = 0; i < 5; i++){
      cout << "Index : " << i <<
        "value : " << name[i];
    }
    //printing null character value
    int value = (int)name[4];
    cout << "value of null chara
      -cter is : " << value;
    return 0;
}
```

Date.....

i) create character array.
ii) input full name.
iii) print the array.

↓

```
int main ()
{
    char ch[100];
    cout << "\n enter name: ";
    cin >> ch;
    cout << "\n Your name is: " <<
                                ch;

    return 0;
}
```

↓ if we enter full name like:-
"Ramesh Kumar"
when we print this array then it will show only :-
"Ramesh"

↓

why??

It is because "cin >> ch;", this line reads the input until it gets the space.
when the space is encountered, it will not read the next character.
So it will read "Ramesh" only.

Date.....

If cin encounters any of these 3, then it will stop reading input:-

- Space (" ")
- Tab ("\t")
- Enter or Newline ("\n")

* How to take full name as an input ??

There is a getline function, that takes full input including spaces also.

```
cin.getline(ch, 50);
```

here we provide max size of the input.

Date.....

Example of getline :-

```
int main()
{
    char name[256], title[256];
    cout << "\n enter name: ";
    cin.getline (name, 256);
    cout << "\n enter your favorite
                movie: ";
    cin.getline (title, 256);
    cout << name << "'s favorite
      movie is " << title ;
    return 0;
}
```

* Parameters in getline function :-

getline(array_name, max_input_size, delimeter);

↓

delimeter means, we explicitly tell that on which character it should stop taking input.

Date.....

Ques Find the length of the string.

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

int getLength(char name[])
{
    int length = 0;
    for(int i = 0; i < 100; i++){
        if(name[i] == '\0'){
            break;
        }
        else{
            length++;
        }
    }
    return length;
}

int main()
{
    char name[100];
    cout << "\n enter name : ";
    cin.getline(name, 100);

    int length = getLength(name);

    cout << "length of your string
        is :" << length;
    return 0;
}
```

Date.....

* Pre-defined function for finding the length of the string is :-

```
strlen(name);
```

↳ include the library "#include<string.h>" to use strlen() length, or we can also use the library "#include<cstring>"

Date.....

Ques Reverse a string. (two pointer approach).

```
#include<iostream>
#include<cstring>
using namespace std;
void reversestring(char name[],
                   int n)
{
    int i=0, j=n-1;
    while(i<=j){
        swap(name[i], name[j]);
        i++;
        j--;
    }
    cout<<"Reverse:"<<name;
}
int main()
{
    char name[100];
    cout<<"enter name:";
    cin.getline(name, 100);
    int n=strlen(name);
    reversestring(name, n);
    return 0;
}
```

Date.....

Ques Replace all spaces, with '@'

```
#include<iostream>
#include<cstring>
using namespace std;
void replaceSpaces(char ch[], int n)
{
    for(int i=0; i<n; i++) {
        if(ch[i] == ' ') {
            ch[i] = '@';
        }
    }
    cout << "Replaced string: " << ch;
}
int main()
{
    char ch[100];
    cout << "Enter name: ";
    cin.getline(ch, 100);
    int n = strlen(ch);
    replaceSpaces(ch, n);
    return 0;
}
```

Ques Palindrome String. (two pointer approach).

```
#include<iostream>
#include<cstring>
using namespace std;
bool palString(char name[], int n)
{
    int i = 0, j = n-1;
    while(i <= j)
    {
        if(name[i] == name[j]){
            i++;
            j--;
        }
        else{
            return false;
        }
    }
    return true;
}
int main()
{
    char name[100];
    cout << "enter name :";
    cin.getline(name, 100);
    int n = strlen(name);
    if(palString(name, n)){
        cout << "palindrome";
    }
    else{
        cout << "not palindrome";
    }
    return 0;
}
```

Teacher's Sign .....

Date.....

Ques convert string in uppercase.

ASCII value of 'a' is 97.
ASCII value of 'A' is 65.

If we want to convert 'e' to 'E', then we can subtract 'a' & add 'A' in any lowercase character. It will convert lowercase to uppercase.
For example, ASCII value of 'e' is 101,

e - a + A
101 - 97 + 65 = 69

this is the ASCII value of 'E'

```
#include<iostream>
#include<cstring>
using namespace std;
void convertUppercase(char name[],
                      int n){
    for(int i=0; i<n; i++){
        name[i] = name[i] - 'a' + 'A';
    }
    cout << name << endl;
}
int main(){
    char name[100];
    cout << "enter name : ";
    cin.getline(name, 100);
    int n= strlen(name);
    convertUppercase(name, n);
    return 0;
}
```

Date.....

Ques convert string in lowercase.

If we subtract 'A' & add 'a' in any uppercase character, then it will convert the character to lowercase.
For example :-

ASCII value of 'E' is 69.
ASCII value of 'A' is 65.
ASCII value of 'a' is 97.

E - A + a
69 - 65 + 97 = 101 → this is the ASCII value of 'e'.

```
void convertLowercase (char name[],
                       int n)
{
    for(int i = 0; i < n; i++){
        name[i] = name[i] - 'A' + 'a';
    }
    cout << name << endl;
}
```

Signal

Teacher's Sign

Strings

Date.....

character array store the sequence of characters.
String also store the sequence of characters.

↓ But both are not same. The difference is that...

In character array, we've created an array in which every variable is of char type.
In Strings, the string itself is a datatype, that the data is of string type.

character array is a data structure of char type values.
String is a datatype itself.

String can also said to be dynamic character array.

To use string, we have to include a library, i.e.,

```
#include <string>
```

* Declare a string:-

```
string str;
```

Date.....

For taking spaces as an input in character array, we use getline function like that :-

```
cin.getline(ch, 50);
```

For taking spaces as an input in character array, we use getline function like this :-

```
getline(cin, str);
```

* Functions used in strings :-

Some of the most important functions that are used in strings are :-

i) size().
ii) empty().
iii) capacity().
iv) push_back()
v) pop_back()
vi) erase(). * very important.
vii) replace() * very important.
viii) find(). ** very important.
ix) substr(). ** very important.
x) compare(). ** very important.
xi) npos.
xii) length().

npos means "no position".

Que: Function used in string.

```
#include<iostream>
#include<string>
using namespace std;
int main()
{
    string str = "Find absence of
          character in string";
    cout << "\n You've entered: " <<
                        str << endl;

    //length of string.
    cout << "Length of your string:"
        << str.length() << endl;

    // check the string is empty or not
    cout << "\n Your string is empty"
        << str.empty() << endl;

    // push_back
    str.push_back('z');
    cout << "After push_back: " << str;

    //pop_back
    str.pop_back();
    cout << "After pop_back: " << str;

    //find substring
    cout << "Substring for 5th index to
    7 character is :" << str.substr(5, 7)
    << endl;
```

Teacher's Sign

Date.....

```
//compare two strings
string A = "Hello world";
string B = "How are you";
if((A.compare(B)) == 0)
{
   cout << "Equal";
}
else
{
   cout << "Not equal";
}

//find in a string.
string str1 = "Hello world";
string str2 = "world";
if(str1.find(str2) != string::npos)
{
   cout << "Index is:" <<
                str1.find(str2);
}
else
{
   cout << "not found";
}

// erase string
string message1 = "this is my
                    message";
cout << "after erase:" <<
          message1.erase(11, 5);
```

Date.....

```
//replace 1 string with another
string message = "this is my
                 first message";
string word = "your";
cout << "after replace: " <<
    message.replace(8, 2, word);

return 0;
}
```

starting index

no. of characters to replace

from which we want to replace