Array of Characters and Strings

Declaration of an array of character

```
char C[10];
```

Initialization of an array of character

```
char Grades[5]={'A','B','C','D','E'};
```

EXAMPLE 1:

```
char Grades[5]={'A','B','C','D','E'};
for (int I=0;I<5;I++)
      cout<<Grades[I]<<endl;
```

OUTPUT

```
A
B
C
D
E
```

EXAMPLE 2:

```
char Grades[5]={'A','B','C','D','E'};
int G,Scores[5]={12,5,10,2,9};//Scores out of 12
for (int I=0;I<13;I++)
{
  G=4-(Scores[I]+2)/3;
  cout<<Scores[I]<<":"<<Grades[G]<<endl;
}
```

OUTPUT

```
12:A
5:C
10:A
2:D
9:B
```

Initialization of an array of characters as a string

```
char City[10]="Delhi";
char University[]="Delhi University";
char Landmark[10]={'T','E','M','P','L','E','\0'};
```

Note: char City[] = {'T','E','M','P','L','E'};
cout << City; // TEMPLE
and if do by using for loop then also we get TEMPLE

EXAMPLE 3:

```
char City[10]="Delhi";
char University[]="Delhi University";
char Landmark[10]={'T','E','M','P','L','E','\0'};
cout<<City<<","<<University<<endl;//Displaying content of Strings
cout<<Landmark<<endl;              //Displaying content of a String
```

OUTPUT

```
Delhi,Delhi University
TEMPLE
```

A **string** is an array of character, which has a null character \0 termination for the content. In the above example, the first two initializations, null character was automatically assigned after the last characters i and y respectively. Whereas, in third initialization, null character is explicitly assigned to make the array of characters to become a string.

EXAMPLE 4

```
char Name[30];
cout<<"Enter Name:";cin>>Name;
cout<<Name<<endl;
```

OUTPUT:

```
Enter Name: Rudra Pratap Singh
Rudra
```

EXAMPLE 4a

```
int A,B,C;
cin>>A>>B>>C;
cout<<A<<":"<<B<<":"<<C<<endl;
```

OUTPUT:

```
120 30 459
120:30:459
```

Underlined Values are the sample inputs from the user

Look at the above EXAMPLE 4, if "Rudra Pratap Singh" is entered by the user for Name variable using cin>>, only the first name "Rudra" will be taken as content of Name and remaining portion of input will be in buffer. Here, **Name** will be a **string** as a null character will be automatically inserted immediately after 'a' of "Rudra". EXAMPLE 4a illustrates clearly that cin>> uses space as a separator for inputs.

EXAMPLE 5

```
char Address[20],City[20],Country[20];
cout<<"Address:";cin>>Address;
cout<<Address<<endl;
cout<<"City:";cin>>City;
cout<<City<<endl;
cout<<"Country:";cin>>Country;
cout<<Country<<endl;
```

OUTPUT:

```
Address:E-40,ABColony Delhi India
E-40,ABColony
City:Delhi
Country:India
```

Underlined Values are the sample inputs from the user

In the EXAMPLE 5, if "E-40,ABColony Delhi India" is entered by the user for Address variable using cin>>, only the first portion "E-40,ABColony" will be taken as content of **Address** and remaining portion of input will be in buffer and will automatically assign **City** as **"Delhi"** and **Country** as "India" from input buffer. The program will not ask the user to give inputs for City and Country explicitly.

EXAMPLE 6

```
char Address[20],City[20],Country[20];
cout<<"Address:";gets(Address);//gets() requires stdio header file
cout<<Address<<endl;
cout<<"City:";gets(City);
cout<<City<<endl;
cout<<"Country:";gets(Country);
cout<<Country<<endl;
```

OUTPUT:

```
Address:E-40,ABColony Delhi India
E-40,ABColony Delhi India
City:Delhi
```

Underlined Values are the sample inputs from the user

```
Delhi
Country:India
India
```

In the EXAMPLE 6, if "E-40,ABColony Delhi India" is entered by the user for Address variable using gets, now the entire content "E-40,ABColony Delhi India" will be taken in Address alongwith spaces. The execution of the program will stop to take inputs for City and Country separately.

Important:

- String Variable can't be assigned with assignment operator =

  i.e.
  ```
  char Name[20];
  Name="Preeti"; //is wrong
  ```
- String Variables can't be compared with relational operators ==, <, >, >=, <=, !=

  i.e.
  ```
  char MyName[20],YorName[20];
  gets(MyName);
  gets(YorName);
  if (MyName==YorName) //is wrong
    cout<<"Both Names are Same"<<endl;
  ```

User-defined Function to copy content from one string to another

```
void COPY(char Another[],char One[])
{
  for (int I=0;One[I]!='\0';I++)
    Another[I]=One[I];
  Another[I]='\0';
}
```

User-defined Function to compare content of one string with another

```
int COMP(char One[],char Two[])
{
  for (int I=0;One[I]==Two[I] &&
               One[I]!='\0' &&
               Two[I]!='\0';I++);
  return One[I]-Two[I];
}
```

User-defined Function to find the number of characters present in a string

```
int LENGTH(char Str[])
{
  for (int I=0;Str[I]!='\0';I++);
    return I;
}
```

User-defined Function to concatenate content of one string with another

```
void CONCAT(char One[],char Two[])
{
  for (int I=0;One[I]!='\0';I++);
  for (int J=0;Two[J]!='\0';J++)
    One[I++]=Two[J];
  One[I]='\0';
}
```

User-defined Function to reverse the content of a string

```
void REVERSE(char Str[])
{
  for (int L=0;Str[L]!='\0';L++);
  for (int J=0;J<L/2;J++)
  {
    char T=Str[J];
    Str[J]=Str[L-J-1];
    Str[L-J-1]=T;
  }
}
```

User-defined Function to extract a portion (substring) of a string

```
void SUBSTR(char Str[],char Sub[],
                    int Pos,int L)
{
  for (int I=0,J=Pos;
        I<L && Str[J]!='\0';J++,I++)
    Sub[I]=Str[J];

  Sub[I]='\0';
}
```

Main() function to illustrate the working of the above string manipulation functions with sample data	OUTPUT

void main()
{
char S[30]="Bharat",T[30]="New Delhi";
char R[30],P[30];
cout<<"Length:"<<LENGTH(S)<<endl;
if (COMP(S,T)==0)
cout<<"Both Strings Same"<<endl;
else
cout<<"Both Strings Different"<<endl;
CONCAT(S,T);
cout<<"Changed String:"<<S<<endl;
COPY(P,T);
REVERSE(T);
cout<<P<<" reversed as "<<T<<endl;
SUBSTR(S,R,6,3);
cout<<R<<" is a Portion of "<<endl;
cout<<S<<endl;
}
	Length:6
	Both Strings Different
	Changed String:BharatNew Delhi
	New Delhi reversed as ihleD weN
	Portion:New is portion of
	BharatNew Delhi

Pre-defined string functions

Header File: string.h

Description	Function	EXAMPLE	OUTPUT
To find length of a string	strlen()	char Str[]="Qutab Minar"; cout<<strlen(Str)<<endl;	11

To copy a string to another	strcpy()	char S[]="Qutab Minar",T[20];
		strcpy(T,S);
		cout<<T<<endl;
			Qutab Minar

To concatenate two strings (to link together, or unite in a series)	strcat()	char S[20]="Qutab",T[]="Minar"; strcat(T,S); cout<<T<<endl<<S<<endl;	Qutab Minar ~~Minar~~ Qutab

To compare two strings
 	
To compare two strings ignoring upper/lowercases
	strcmp()
	
	strcmpi()
		char
		S[20]="Qutab",T[20]="Minar";
		if (strcmp(S,T)==0)
		cout<<"Same"<<endl
		else
		cout<<"Different"<<endl;
			Different

To reverse content of a string	strrev()	char S[20]="Qutab"; cout<<strrev(S)<<endl; cout<<S<<endl;	batuQ batuQ
To convert content of a string into uppercase	strupr()	char S[20]="Qutab"; cout<<strupr(S)<<endl; cout<<S<<endl;	QUTAB QUTAB
To convert content of a string into lowercase	strlwr()	char S[20]="Qutab"; cout<<strlwr(S)<<endl; cout<<S<<endl;	qutab qutab

Array of Strings

Initialization of an array of Strings

```
char Cities[3][20]={"Delhi","Kolkata","Mumbai"};
for (int I=0;I<3;I++)
  cout<<Cities[I]<<":";
cout<<endl;
```

OUTPUT

```
Delhi:Kolkata:Mumbai:
```

User Defined function to allow user to enter the content in an array of String	User Defined function to display the content in an array of String

`void Enter(char S[][20],int N)`
`{`
`  for (int I=0;I<N;I++)`
`  {`
`  cout<<"String["<<I<<"]?";`
`  gets(S[I]);`
`  }`
`}`
	`void Display(char S[][20],int N)`
	`{`
	`  for (int I=0;I<N;I++)`
	`  cout<<"["<<I<<"]"<<S[I]<<endl;`
	`}`

User Defined function to search for a string content from an array of String	**User Defined function to sort an array of String in ascending order**

`void Search(char S[][20],int N,`
`  char SS[])`
`{`
`  int Found=0;`
`  for (int I=0;I<N;I++)`
`  if (strcmp(S[I],SS)==0)`
`  {`
`  cout<<"Found..."<<endl;`
`  Found++;`
`  }`
`  if (Found==0)`
`  cout<<"Not Found..."<<endl;`
`}`
	`void Sort(char S[][20],int N)`
	`{`
	`  for (int I=0;I<N-1;I++)`
	`  for (int J=0;J<N-I-1;J++)`
	`  if (strcmp(S[J],S[J+1])>0)//1`
	`  {`
	`  char T[20];`
	`  strcpy(T,S[J]);`
	`  strcpy(S[J],S[J+1]);`
	`  strcpy(S[J+1],T);`
	`  }`
	`}`
	`//Statement marked with 1 will be`
	`//changed to the following for`
	`//descending order sorting`
	`//  if(strcmp(S[J],S[J+1])<0)`

main() function to illustrate the working of the above array of string manipulation functions with sample data	OUTPUT

`void main()`
`{`
`  char  Cities[3][20]={"Kolkata","Delhi","Mumbai"};`
`  Display(Cities,3);`
`  Search(Cities,3,"Mumbai");`
`  Search(Cities,3,"Chennai");`
`  Sort(Cities,3);`
`  Display(Cities,3);`
`}`
	`[0]Kolkata`
	`[1]Delhi`
	`[2]Mumbai`
	`Found...`
	`Not Found...`
	`[0]Delhi`
	`[1]Kolkata`
	`[2]Mumbai`