```
{
char str[20];
char str1[20];
int opt,len;
printf("\n  MAIN MENU");
printf("\n 1. Convert string into upper case");
printf("\n 2. Reverse the string");
printf("\n 3. Copy one string into another string");
printf("\n  4.Compute length of string ");
printf("Enter string ");
scanf("%s", &str);
printf("Enter your choice");
scanf("%d",&opt);
switch(opt)
{
case 1:                 strupr(str);
```



```
                        printf("The string in uppercase is :%s ",str);
                        break;

case 2:                 strrev(str);
                        printf("The reverse of string is : %s",str);
                        break;
case 3:                 strcpy(str1,str);
                        printf("New copied string is : %s",str1);
                        break;
case 4:                 len=strlen(str);
                        printf("The length of the string is :  %s",len);
                        break;
default:                printf("Ypu have entered a wrong choice.");
}
```

ARRAYS:

What is an array?

An array is a collection of similar datatype that are used to allocate memory in a sequential manner.

Syntax : <data type> <array name>[<size of an array>]

Subscript or indexing: A subscript is property of an array that distinguishes all its stored elements because all the elements in an array having the same name (i.e. the array name). so to distinguish these, we use subscripting or indexing option.

e.g. int ar[20];

First element will be: int ar[0];

Second element will be: int ar[1];

Third element will be: int ar[2];

Fourth element will be: int ar[3];

Fifth element will be: int ar[4];

Sixth element will be: int ar[5];

So on.....

Last element will be: int ar[19];

- NOTE: An array always starts from 0 indexing.
- Example: int ar[20];

This above array will store 20 integer type values from 0 to 19.

Advantage of an array:

- Multiple elements are stored under a single unit.
- Searching is fast because all the elements are stored in a sequence.

Types of Array

1. Static Array
2. Dynamic Array.

Static Array

An array with fixed size is said to be a static array.

Types of static array:

1. One Dimensional Array
2. Two Dimensional Array.
3. Multi Dimensional Array.

1. One Dimensional Array

An Array of elements is called 1 dimensional, which stores data in column or row form.

Example: int ar[5];

This above array is called one dimensional array because it will store all the elements in column or in row form

2. Two Dimensional Array.

An array of an array is said to be 2 dimensional array , which stores data in column androw form

Example: int ar[4][5];

This above array is called two dimensional array because it will store all the elements in column and in row form

NOTE: In above example of two dimensional array, we have 4 rows and 5 columns.

NOTE: In above example of two dimensional array, we have total of 20 elements.

3. Multi Dimensional Array.

This array does not exist in c and c++.

Dynamic Array.

This type of array also does not exist in c and c++.

Example: Program based upon array:

WAP to store marks in 5 subjects for a student. Display marks in 2nd and 5thsubject.

```
#include<stdio.h>
#include<conio.h>
void main()
{
int ar[5];
int i;
for(i=0;i<5;i++)
{
printf(" \n Enter marks in ",i, "subject");
scanf("%d",&ar[i]);
}
printf("Marks in 2nd subject is: ",ar[1]);
printf("Marks in 5th subject is: ",ar[4]);

}
```

STRINGS

What is String?

- A string is a collection of characters.
- A string is also called as an array of characters.
- A String must access by %s access specifier in c and c++.
- A string is always terminated with \0 (Null) character.
- Example of string: "Gaurav"
- A string always recognized in double quotes.
- A string also consider space as a character.

- Example: " Gaurav Arora"
- The above string contains 12 characters.
- Example: Char ar[20]
- The above example will store 19 character with I null character.

Example: Program based upon String.

WAP to accept a complete string (first name and last name) and display hello message in the output.

```
# include<stdio.h>
#include<conio.h>
#include<string.h>
void main ()
{
char str1[20];
char str2[20];
printf("Enter First Name");
scanf("%s",&str1);
printf("Enter last Name");
scanf("%s",&str2);
puts(str1);
puts(str2);
}
```

String Functions in C:

Our c language provides us lot of string functions for manipulating the string.

All the string functions are available in string.h header file.

These String functions are:

1. strlen().
2. strupr().
3. strlwr().
4. strcmp().

5. strcat().
6. strapy().
7. strrev().

1. strlen().

This string function is basically used for the purpose of computing the ength of string.

Example: char str="Gaurav Arora";

```
int length= strlen(str);
printf("The length of the string is =",str);
```

2. strupr().

This string function is basically used for the purpose of converting the case sensitiveness of the string i.e. it converts string case sensitiveness into uppercase.

Example: char str = "gaurav"

```
strupr(str);
printf("The uppercase of the string is : %s",str);
```

3. strlwr ().

This string function is basically used for the purpose of converting the case sensitiveness of the string i.e it converts string case sensitiveness into lowercase.

Example: char str = "gaurav"

```
strlwr(str);
printf("The Lowercase of the string is :%s ",str);
```

4. strcmp ().

This string function is basically used for the purpose of comparing two string.

This string function compares two strings character by characters.

Thus it gives result in three cases:

Case 1: if first string > than second string then, result will be true.

Case 2: if first string < than second string then, result will be false.

Case 3: if first string = = to second string then, result will be zero.

Example:

```
char str1= "Gaurav";
char str2= "Arora";
char str3=strcmp(str1,str2);
printf("%s",str3);
```

5. strcat().

This string function is used for the purpose of concatenating two strings ie.(merging two or more strings)

Example:

```
char str1 = "Gaurav";
char str2 = "Arora";
char str3[30];
str3=strcat(str1,str2);
printf("%s",str3);
```

6. strcpy()

This string function is basically used for the purpose of copying one string into another string.

```
char str1= "Gaurav";
char str2[20];
str2 = strcpy(str2,str1);
printf("%s",str2);
```

6. strrev()

This string function is basically used for the purpose of reversing the string.

```
char str1= "Gaurav";
char str2[20];
```

```
str2= strrev(str2,str1);
printf("%s",str2);
```

Example: Program based upon string functions.

WAP to accept a string and perform various operations:

1. To convert string into upper case.
2. To reverse the string .
3. To copy string into another string.
4. To compute length depending upon user choice.

```
# include<stdio.h>
# include<conio.h>
#include<string.h>

void main()
{
char str[20];
char str1[20];
int opt,len;
printf("\n  MAIN MENU");
printf("\n 1. Convert string into upper case");
printf("\n 2. Reverse the string");
printf("\n 3. Copy one string into another string");
printf("\n  4.Compute length of string ");
printf("Enter string ");
scanf("%s", &str);
printf("Enter your choice");
scanf("%d",&opt);
switch(opt)
{
case 1:                         strupr(str);
```

```
                printf("The string in uppercase is :%s ",str);
                break;

case 2:         strrev(str);
                printf("The reverse of string is : %s",str);
                break;
case 3:         strcpy(str1,str);
                printf("New copied string is : %s",str1);
                break;
case 4:         len=strlen(str);
                printf("The length of the string is :  %s",len);
                break;
default:        printf("Ypu have entered a wrong choice.");
}
```