

EXPERIMENTS

- 11. Sort an Array in Ascending Order.*
- 12. Evaluate the Length of a String.*
- 13. Copy Contents of a String to Another.*
- 14. Compare two strings.*
- 15. Concatenate two strings.*
- 16. Reverse a Given String.*
- 17. Check a Palindrome String.*

Expt 11: Sort an Array in Ascending Order.

Aim:

To sort an array of N integers in ascending order.

Algorithm:

- Sort N integers in an array
- First element taken, compared with all other elements of the array, the process is continued till the last element, in a loop.
- If the remainder is not 0, increment the counter
- End the loop
- Display the counter as the result

```

//***** //
Program name : Ascend.c
// Author : Anantha Krishnan R J
// Date Written : 15/06/2021
// Date Compiled : 15/06/2021
// Aim of the Program : To sort an array of N integers in ascending order.
//*****
//*****
#include<stdio.h>

int main()
{
    int n, i, a[100], temp, j;

    printf("Enter the number of integers to be arranged:\n");
    scanf("%d", &n);

    printf("Enter the numbers\n");
    for(i=0; i<n; i++)
    {
        scanf("%d", &a[i]);
    }
    for(i=0; i<(n-1); i++)
    {
        for(j=0; j<(n-1-i); j++)
        {
            if(a[j]>a[j+1])
            {
                temp=a[j];
                a[j]=a[j+1];
                a[j+1]=temp;
            }
        }
    }
}

```

```
printf("The numbers in ascending order are:\n");
```

```
for(i=0; i<n; i++)
```

```
    printf("%d\n", a[i]);
```

```
}
```

Output :

```
/tmp/ozzDcb52eT.o
Enter the number of integers to be arranged:
5
Enter the numbers
10
5
7
1
100
The numbers in ascending order are:
1
5
7
10
100
```

Expt 12: Evaluate the Length of a String

Aim:

To find the length of a given string.

Algorithm:

- Read the string through the keyboard.
- Use strlen function to find the length of the string.
- Display the result of length of string.

Program:

```
/**
Program name : stringlen.c
// Author : Anantha Krishnan R J
// Date Written : 15/06/2021
// Date Compiled : 15/06/2021
// Aim of the Program : To find the length of a given string.
**/

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

int main()
{
    char str[100];
    int len;
    printf("Enter the string whose length is to be calculated:\n");
    scanf("%s", str);
    len=strlen(str);
    printf("The length of the entered string is = %d",len);
}
```

Output:

```
Enter the string whose length is to be calculated:
helloooooo
The length of the entered string is = 10|
```

Expt 13: Copy Contents of a String to Another.

Aim:

To copy the contents of a string to another without using library functions.

Algorithm:

- Declare 2 array of the same length.
- Use a loop and copy each character of the source string to the destination string.
- Display both the strings.

```

//*****
Program name : copyst.c
// Author : Anantha Krishnan R J
// Date Written : 15/06/2021
// Date Compiled : 15/06/2021
// Aim of the Program : To copy the contents of a string to another without using library
functions.
//*****
//*****

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

int main()
{
    int i, len;
    char a[100], b[100], temp;

    printf("Enter the string to be copied:\n");
    fgets(a, 100, stdin);

    len=strlen(a);

    for(i=0; i<len; i++)
    {
        temp=a[i];
        a[i]=b[i];
        b[i]=temp;
    }
    printf("Copied string is:\n %s", b);
}

```

Output:

```
Enter the string to be copied:
```

```
coding
```

```
Copied string is:
```

```
coding
```

```
|
```

Expt 14: Compare Two Strings

Aim:

To compare two strings without any standard functions and if the strings are not identical, display the position where the characters are different.

Algorithm:

- Read in the two strings.
- Using a loop, check the corresponding characters of both the strings.
- If identical, the loop is continued.
- Otherwise, a counter variable 'diff' is incremented.
- Display the locations where the strings are non-identical.
- Display the number of places where the two strings are different.

```

//***** //
Program name : cmpstr.c
// Author : Anantha Krishnan R J
// Date Written : 15/06/2021
// Date Compiled : 15/06/2021
// Aim of the Program : To compare two strings without any standard functions and if the
strings are not identical, display the position where the characters are different.

//*****
//*****

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

int main()
{
    int len, i, diff=0;
    char a[100], b[100];

    printf("Enter the first string:\n");
    fgets(a, 100, stdin);

    printf("Enter the second string:\n");
    fgets(b, 100, stdin);

    len=strlen(a);
    for(i=0; i<len; i++)

        if(a[i]!=b[i])
```

```

    {
        continue;
    }
    else
    {
        s++;
        printf("The point of difference is %d\n", i+1);
    }

if(s==0)
{
    printf("The strings are same.\n");
}
else
{
    printf("The strings are different.\n");
}

printf("The number of characters that are different : %d\n", diff);

}

```

Output:

```

Enter the first string:
hello
Enter the second string:
world
The point of difference is 1
The point of difference is 2
The point of difference is 3
The point of difference is 5
The strings are different.
The number of characters that are different :4

```

```

Enter the first string:
hello
Enter the second string:
hello
The strings are same.
The number of characters that are different :0
|

```


Expt 15: Concatenate Two Strings

Aim:

To concatenate two strings without a standard function.

Algorithm:

- Read the two strings.
- Declare a third string to store the concatenated string.
- Use a loop, store the first string to the third.
- Store the 2nd string to the 3rd, starting from the current position.
- Display the 3rd string.

```
/**
Program name : concatstr.c
// Author : Anantha Krishnan R J
// Date Written : 15/06/2021
// Date Compiled : 15/06/2021
// Aim of the Program : To concatenate two strings.
**/

#include<stdio.h>

#include<string.h>

int main()
{
    int i=0, j=0;
    char a[100], b[100], c[100];

    printf("Enter the first string:\n");
    scanf("%s", a);

    printf("Enter the second string:\n");
    scanf("%s", b);

    while(a[i]!='\0')
    {
        c[i]=a[i];
        i++;
    }
    while(b[j]!='\0')
    {
        c[i]=b[j];
```

```
        i++;  
        j++;  
    }  
  
    c[i]='\0';  
    printf("The concatenated string is:\n%s", c);  
  
}
```

Output:

```
Enter the first string:  
Hello  
Enter the second string:  
World  
The concatenated string is:  
HelloWorld
```

Expt 15: Reverse a Given String

Aim:

To reverse a given string.

Algorithm:

- Read in a string 'str'
- Assign a variable 'len' to the string length
- Use loop len/2 times
- Assign a temporary variable 'temp'
- Make the input string equal to the temp string and then assign temp string to the reverse string
- Print out str as the reverse string

```
/**/
```

```
Program name : revstr.c
```

```
// Author : Anantha Krishnan R J
```

```
// Date Written : 15/06/2021
```

```
// Date Compiled : 15/06/2021
```

```
// Aim of the Program : To reverse a given string.
```

```
/**/
```

```
/**/
```

```
int main()
```

```
{
```

```
    int i=0, len;
```

```
    char str[100], temp[100];
```

```
    printf("Enter the string:\n");
```

```
    scanf("%s", str);
```

```
    len=strlen(str);
```

```
    for(i=0; i<len/2; i++)
```

```
    {
```

```
        temp[i]=str[i];
```

```
        str[i]=str[len-1-i];
```

```
        str[len-1-i]=temp[i];
```

```
    }
```

```
    printf("The reversed string is:\n%s", str);
```

```
}
```

Output:

```
Enter the string:  
PROGRAMMING  
The reversed string is:  
GNIMMARGORP|
```

Expt 17: Check a Palindrome String

Aim:

To check whether the given string is a palindrome or not.

Algorithm:

- Read in a string 'str'
- Assign a variable len, which gives the length of a string
- Using loop, assign a variable i=0, and check if the first and last characters are the same
- Increment i
- If they are found equal, display that it is a palindrome
- If found unequal, display that it is not a palindrome

```
//***** //
```

```
Program name : palinchk.c
```

```
// Author : Anantha Krishnan R J
```

```
// Date Written : 15/06/2021
```

```
// Date Compiled : 15/06/2021
```

```
// Aim of the Program : To check whether the given string is a palindrome or not.
```

```
//*****
```

```
//*****
```

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

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

```
int main()
```

```
{
```

```
    int i=0, len, s=0;
```

```
    char str[100];
```

```
    printf("Enter the string:\n");
```

```
    scanf("%s", str);
```

```
    len=strlen(str);
```

```
    for(i=0; i<len/2; i++)
```

```
    {
```

```
        if(str[i]==str[len-i-1])
```

```
        {
```

```
            continue;
```

```
        }
```

```
    else
```

```
        {  
            s++;  
        }  
    }  
    if(s==0)  
    {  
        printf("It is a palindrome.");  
    }  
    else  
    {  
        printf("It is not a palindrome.");  
    }  
}
```

Output:

```
Enter the string:  
Hellloooo  
It is not a palindrome.
```

```
Enter the string:  
ROTOR  
It is a palindrome.|
```