AIM: Reverse a String in C

DESCRIPTION:

Reversing a string is the technique that reverses or changes the order of a given string so that the last character of the string becomes the first character of the string and so on. Furthermore, we can also check the Palindrome of the given string by reversing the original string.

Recursion function: A recursion function is a function that continuously calls the same function without using a looping statement.

For example, we enter a string "**APPLE**", and then use the reverse algorithm. The reverse algorithm returns the string "**ELPPA**" which is completely reverse of the original string.

A	P	P	L	E

↓ **REVERSE**

E	L	P	P	A

Different ways to find the reverse of a string in the C

Following are the various ways to find the reverse of a string in the C programming language:

1. Reverse a string using the strrev() function
2. Reverse a string without using the library function
3. Reverse a string using the recursion function
4. Reverse a string using for loop
5. Reverse a string using while loop
6. Reverse a string using pointers
7. Reverse a string to check for Palindrome

Program 1: Print the reverse of a string using strrev() function

Program1.c

```
#include <stdio.h>
#include <string.h>
int main()
{
   char str[40];  / declare the size of character string
   printf (" \n Enter a string to be reversed: ");
```

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

    // use strrev() function to reverse a string
    printf (" \n After the reverse of a string: %s ", strrev(str));
    return 0;
}
```

Output:

```
Enter a string to be reversed: AMBULANCE
After the reverse of a string: ECNALUBMA
```

Program 2: Print the reverse of a string without using the library function

Program2.c

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

// function definition of the revstr()
void revstr(char *str1)
{
    // declare variable
   int i, len, temp;
   len = strlen(str1); // use strlen() to get the length of str string

   // use for loop to iterate the string
   for (i = 0; i < len/2; i++)
    {
        // temp variable use to temporary hold the string
        temp = str1[i];
        str1[i] = str1[len - i - 1];
        str1[len - i - 1] = temp;
    }
}

    int main()
    {
        char str[50]; // size of char string
        printf (" Enter the string: ");
        gets(str); // use gets() function to take string

        printf (" \n Before reversing the string: %s \n", str);

// call revstr() function
        revstr(str);
        printf (" After reversing the string: %s", str);
    }
```

Output

```
Enter the string: Welcome Friends
Before reversing the string: Welcome Friends
After reversing the string: sdneirF emocleW
```

Program 3: Print the reverse of a string using the recursion function

Program3

```
#include <stdio.h>

#include <string.h>

// use recursion function
void revstr(char *str1)
{
    // declare static variable
    static int i, len, temp;
    len = strlen(str1); // use strlen() to get the length of str string


    if (i < len/2){
// temp variable use to temporary hold the string
        temp = str1[i];
        str1[i] = str1[len - i - 1];
        str1[len - i - 1] = temp;
        i++;
        revstr(str1); // recusively calls the revstr() function
    }
}

int main()
{
        char str1[50]; // size of char string
        printf (" Enter the string: ");
        gets(str1); // use gets() function to take string

        printf (" \n Before reversing the string: %s \n", str1);

        // call revstr() function
        revstr(str1);
        printf (" After reversing the string: %s", str1);
}
```

Output

```
Enter the string: LIFE INSURANCE

 Before reversing the string: LIFE INSURANCE
 After reversing the string: ECNARUSNI EFIL
```

Program 4: Print the reverse of a string using for loop

Program4

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
void main()
{
    char str[50], temp; // define the size of str[] array
    int i, left, right, len;
    printf (" \n Display a reverse string in the C: \n");
    printf (" ---------------------- ");
    printf (" \n Enter a string to reverse order: ");
    scanf( "%s", &str);
    len = strlen(str); // get the length of the string
    left = 0; // set left index at 0
    right = len - 1; // set right index len - 1
    // use for loop to store the reverse string
   for (i = left; i <right; i++)
   {
        temp = str[i];
        str[i] = str[right];
        str[right] = temp;
        right--;
    }
    printf (" The reverse of the original string is: %s ",  str);
    getch();
}
```

Output

```
Display a reverse string in the C:
 -----------------------
 Enter a string to reverse order: APPLE
 The reverse of the original string is: ELPPA
```

Program 5: Print the reverse of a string using while loop

Program5.c

```
#include <stdio.h>
#include <string.h>
int main()
```

```
{
char str1[50], temp; // declare and initialize the size of string array.
Int i = 0, j =0;
printf (" Enter a string to be reversed: ");
scanf( "%s", str1);
j = strlen (str1) - 1; // get the length of the string
// use while loop to define the condition
while ( i < j )
{
// use temp variable to store the characters of str1
temp = str1[j];
str1[j] = str1[i];
str1[i] = temp;
i++; // i incremented by 1
j--; // j is decremented by 1
}
printf (" The reversed of the string: %s", str1);
return 0;
}
```

Output

```
Enter a string to be reversed: JAVATPOINT
 The reversed of the string: TNIOPTAVAJ
```

Program 6: Print the reverse of a string using pointers

Program6.c

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

int str_len( char *st);
void revstr( char *st);
int main()
{
    char st[50];
    printf (" Enter a string to be reversed: ");
    scanf( "%s", st);

    revstr(st);

    printf (" The reverse string is: %s", st);
    return 0;
}

void revstr (char *st)
{
    in len, i;
    char *start, *end, temp;

    len = str_len (st);
    start = st;
    end = st;

    for (i = 0; i < len - 1; i++)
    end++;

    for(i = 0; i < len/2; i++)
    {
        temp = *end;
        *end = *start;
       *start = temp;

        start++;
        end--;
    }
}

int str_len (char *ptr)
{
    int i = 0;
    while ( *(ptr + i) != '\0')
    i++;
    return i;
}
```

Output

```
Enter a string to be reversed: JAVATPOINT
 The reverse string is: TNIOPTAVAJ
```

Program 7: Program to check whether the reverse string is a Palindrome

Program7.c

```
#include <stdio.h>
#include <string.h>
int main ()
{
    // declare variables
    char str1[30];
    int i, len, flag = 0;

    printf (" Enter a string: ");
    scanf ("%s", str1);
```

```
  len = strlen( str1 ); // get the string length


 for ( i = 0; i < len; i++)
  {
      // str1[i] is not equal to str1[len-i-1]
      if (str1[i] != str1[len - i - 1])
  {
          flag = 1;
          break; // exit from if statement
    }
  }
  if (flag)
  {
       printf (" %s is not a palindrome string", str1);
  }
  else
  {
       printf (" %s is a palindrome", str1);
  }
  return 0;
}
```

Output

```
Enter a string: madam
madam is a palindrome string.
```