

**6c.** Write a C program that takes a string as input from the user, reverses it in place, and then prints the reversed string.

### **Description:**

This program reads a string from the user and reverses it in place without using any built-in string functions like `strrev()`.

- The program stores the string in a character array with a fixed size.
- It then uses a two-pointer approach:
- One pointer starts at the beginning (left), and the other at the end (right).
- The characters at these positions are swapped iteratively until they meet in the middle.
- Finally, the program prints the reversed string.

Example:

Enter a string: hello

Reversed string: olleh

### **Algorithm:**

Step 1: Start

Step 2: Declare a character array `str` to store the string and two integer variables `left` and `right`.

Step 3: Prompt the user to enter a string and store it in `str`.

Step 4: Initialize `left = 0` (pointing to the first character) and `right = length of the string - 1` (pointing to the last character).

Step 5: Loop until `left` is less than `right`

- Swap `str[left]` and `str[right]`
- Increment `left` and decrement `right`

Step 6: When `left` meets `right`, the string is fully reversed.

Step 7: Print the reversed string.

Step 8: Stop.

### **Source Code:**

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

int main() {
```

```
char str[100];
int i, len;
char temp;

// Taking input from the user
printf("Enter a string: ");
scanf("%s", str);

len = strlen(str); // Get string length

// In-place reversal
for (i = 0; i < len / 2; i++) {
    temp = str[i];
    str[i] = str[len - i - 1];
    str[len - i - 1] = temp;
}

// Displaying the reversed string
printf("Reversed string: %s\n", str);

return 0;
}
```

### Sample Output:

```
Enter a string: Hello
Reversed string: olleH
```

```
Enter a string: Malayalam
Reversed string: malayalaM
```