Vishwakarma Institute Of Information Technology

Department Of Computer Engineering

- **Name** : Abhinav Belhekar
- **Roll No.** : 224033
- **PRN** : 22010389
- **Division** : SY – D (Computer Department)
- **Batch** : D2
- **Subject** : Fundamentals Of Data Structure

Assignment - 01

❖ **Aim :** To perform following string operations with & without pointer
- 1) Length 2) Copy 3) Concatenation 4) Compare 5) Reverse

❖ **Algorithm :**

Function	Without Pointers	With Pointer
Length	1. Declare & accept string 2. Initialize counter i and set it to 0. 3. Check whether element at **pointed by the counter** is NULL or '\0' or not 4. If yes then increment counter & go to 3 5. If no then stop the loop and return counter value. O(n) : n = length of string	1. Declare & accept string 2. Initialize counter i and set it to 0. 3. Check if pointer is NULL ? 4. If no then increment i & the pointer 5. If yes then stop the loop & return the counter value O(n) : n = length of string
Copy	1. Declare & accept string source & destination. 2. Declare a counter 'i' & initialize it to 0. 3. Check whether element pointed by the counter is NULL ? 4. If not then make i^{th} element of destination string to i^{th} element of source string 5. If yes then string is copied. O(n): n = length of string	1. Declare & accept string source & destination. 2. Check whether element pointed by source is NULL ? 3. If not then copy the element to destination, increment source & destination counter 4. If yes then string is copied. O(n): n = length of string
Compare	1. Declare & accept string s1, s2. 2. Return difference between length of s1, s2 by calling the length function already declared O(n1+n2): n1, n2 = length of s1, s2.	1. Declare & accept string s1, s2. 2. Return difference between length of s1, s2 by calling the length function already declared O(n1+n2): n1, n2 = length of s1, s2.

Function	Without Pointers	With Pointer
Reverse	1. Declare & accept string 2. Declare a i, j and initialize i = 0 & j to length of string - 1. 3. Is i<j ? 4. If yes then replace i^{th} element of string with j^{th} element, increment i, decrease j. 5. If no then exit the loop. O(1.5n): n = length of given string	1. Declare & accept string. 2. Declare a temporary string & initialize it to string. 3. Move the temporary pointer to end. 4. Is string < temporary? 5. If yes then swap string pointer with temporary, decrement temporary, increment string. 6. If not then exit the loop. O(1.5n): n = length of given string
Concatenation	1. Declare & accept string s1, s2. 2. Declare counters 'i', 'j' & initialize 'l' to length of first string, 'j' to zero. 3. Check whether element pointed by counter 'j' is NULL or '\0' or not. 4. If not then put the j^{th} element of second string to $(i+j)^{th}$ position of first string, then increment the counter. 5. If yes then stop. O(n1+n2): n1, n2 = length of s1, s2.	1. Declare & accept string s1, s2. 2. Move s1 pointer to the end, 3. Is s2 NULL ? 4. If not then copy s2 to s1, increment s1, s2 5. If yes then exit the loop. 6. Add termination character at the end. O(n1+n2): n1, n2 = length of s1, s2.

❖ **Code:**

main.cpp

```
/*
* @CoderAbhinav
* @bried Assignment 1 Part 1 : Write string functions
* @author Abhinav Belhekar
* @Roll No : 224033
* @Batch : D2
* @Division : SY - D (Comp)
* @date
*/

/*
Perform following string operations with and without pointers:
1. Length
2. Copy
3. Concat
4. Compare
5. Reverse.
*/

#include <iostream>
#include "operations/pointer.h"
#include "operations/nopointer.h"

void UI();

int main()
{
```

```
    UI();
    return 0;
}



void UI()
{
    pointer ptr;
    nopointer noptr;
    std::cout << "Welcome, Presenting you simple string\n";
    char* s = new char;
    char* d = new char;
    char* e = new char;
    int sel = -1;
    while (sel != 0)
    {
        std::cout << "\n---------------------------------";
        std::cout << "\n1 -> Get Length Of Given String";
        std::cout << "\n2 -> Copy string to another array";
        std::cout << "\n3 -> Concatonate two strings";
        std::cout << "\n4 -> Compare two strings";
        std::cout << "\n5 -> Reverse given string";
        std::cout << "\n0 -> Exit the program.";
        std::cout << "\n> ";
        std::cin >> sel;
        std::cout << "You Selected " << sel << "\n";
        if (sel == 1)
        {
            std::cout << "\nEnter First string : ";
            std::cin.ignore();
            std::cin >> s;
            std::cout << "\nThe length of string (With Pointer) :" << s << " is =
" << ptr.getLength_ptr(s);
            std::cout << "\nThe length of string (Without Pointer) :" << s << "
is = " << noptr.getLength(s);
        }
        else if (sel == 2)
        {
            std::cout << "Enter String to Copy : ";
            std::cin.ignore();
            std::cin >> s;
            ptr.strcpy_ptr(s, d);
            std::cout << "The given string copied to array d\n";
            std::cout << "Here is the copied string (with pointer): " << d;
            noptr.strcpy(s,e);
            std::cout << "\nThe given string copied to array e\n";
            std::cout << "Here is the copied string (without pointer): " << d;
        }
        else if (sel == 3)
        {
            std::cout << "\nEnter First string : ";
            std::cin.ignore();
            std::cin >> s;
            char* temp = new char;
            noptr.strcpy(s, temp);
            std::cout << "\nEnter Second string : ";
            std::cin.ignore();
            std::cin >> d;
            ptr.add_ptr(s, d);
            noptr.add(temp, d);
            std::cout << "\nConcatonated string is (with pointer): " << s;
            std::cout << "\nConcatonated string is (without pointer): " << temp;
        }
        else if (sel == 4)
        {
```

```
            std::cout << "\nEnter First string : ";
            std::cin.ignore();
            std::cin >> s;
            std::cout << "\nEnter Second string : ";
            std::cin.ignore();
            std::cin >> d;
            int diff = noptr.compare(s, d);
            if (diff > 0)
            {
                std::cout << "\nFirst string has more character than second
one.";
            }
            else if (diff < 0)
            {
                std::cout << "\nSecond string has more character than the first
one.";
            }
            else
            {
                std::cout << "\nBoth strings are equal.";
            }
            std::cout << "Difference of character is : " << abs(diff);
        }
        else if (sel == 5)
        {
            std::cout << "Enter String to Reverse : ";
            std::cin.ignore();
            std::cin >> s;
            char* temp = new char;
            noptr.strcpy(s, temp);
            ptr.reverse_ptr(s);
            noptr.reverse(temp);
            std::cout << "\nReversed string is (with pointer): " << s;
            std::cout << "\nReversed string is (with pointer): " << temp;
        }

    }
    std::cout << "Exited!";
}
```


operations/nopointers.h

```
#include <iostream>

class nopointer
{
public:
    int getLength(char *);
    void strcpy(char *, char *);
    int compare(char *, char *);
    void reverse(char *);
    void add(char *, char *);
};

// return length of string
// O(n) -> n : size of string
int nopointer::getLength(char *s)
{
    int i = 0;
    while (s[i])
    {
        i++;
    }
    return i;
}
```

```
// copy src to des
// O(n) -> n : size of src
void nopointer::strcpy(char *src, char *des)
{
    int i = 0;
    while (src[i])
    {
        des[i] = src[i];
        i++;
    }
    des[i] = '\0'; // adding a terminating character so we don't get the garbage
values
}

// returns difference between s1, s2
// O(n1 + n2) -> n1, n2 : lengths of s1, s2
int nopointer::compare(char *s1, char *s2)
{
    return getLength(s1) - getLength(s2);
}


// reverse given string
// O(1.5n) -> n : length of string
void nopointer::reverse(char *s)
{
    int high = getLength(s) - 1;
    int low = 0;
    while (low < high)
    {
        char temp = s[high];
        s[high] = s[low];
        s[low] = temp;
        high--;
        low++;
    }
}


// adds s2 at the end of s1
// O(n1, n2) -> n1, n2 : length of s1, s2
void nopointer::add(char *s1, char *s2)
{
    int i = 0;
    int dis = getLength(s1);
    while (s2[i])
    {
        s1[dis + i] = s2[i];
        i++;
    }
    s1[dis + i] = '\0';
    // adding a terminating character so we don't get the garbage values
}
```

Operations/pointer.h

```
#include <iostream>

class pointer
{
public:
    int getLength_ptr(char *);
    void strcpy_ptr(char *, char *);
    int compare_ptr(char *, char *);
    void reverse_ptr(char *);
    void add_ptr(char *, char *);
```

```
};

// return length of string
// O(n) -> n : size of string
int pointer::getLength_ptr(char *s)
{
    int i = 0;
    while (*s)
    {
        i++;
        s++;
    }
    return i;
}

// copy src to des
// O(n) -> n : size of src
void pointer:: strcpy_ptr(char *src, char *des)
{
    while (*src)
    {
        *des = *src;
        des++;
        src++;
    }
    *des = '\0'; // adding a terminating character so we don't get the garbage
values
}

// returns difference between s1, s2
// O(n1 + n2) -> n1, n2 : lengths of s1, s2
int pointer::compare_ptr(char *s1, char *s2)
{
    return getLength_ptr(s1) - getLength_ptr(s2);
}


// reverse given string
// O(1.5n) -> n : length of string
void pointer:: reverse_ptr(char *s)
{
    char *end = s;
    while (*(end + 1))
    {
        end++;
    }
    while (s < end)
    {
        char temp = *s;
        *s = *end;
        *end = temp;
        s++;
        end--;
    }
}


// adds s2 at the end of s1
// O(n1, n2) -> n1, n2 : length of s1, s2
void pointer:: add_ptr(char *s1, char *s2)
{
    int i = 0;
    while (*s1)
    {
        s1++;
    }
    while (*s2)
```

```
    {
        *s1 = *s2;
        s1++;
        s2++;
    }
    *s1 = '\0';
    // adding a terminating character so we don't get the garbage values
}
```

❖ Output:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS G:\viit notes 2020-21\Sem III SY\Fundamentals Of Data
Structure\Practical\Assignment 1> ./main
Welcome, Presenting you simple string

--------------------------------
1 -> Get Length Of Given String
2 -> Copy string to another array
3 -> Concatonate two strings
4 -> Compare two strings
5 -> Reverse given string
0 -> Exit the program.
> 1
You Selected 1

Enter First string : Abhinav

The length of string (With Pointer) :Abhinav is = 7
The length of string (Without Pointer) :Abhinav is = 7

--------------------------------
1 -> Get Length Of Given String
2 -> Copy string to another array
3 -> Concatonate two strings
4 -> Compare two strings
5 -> Reverse given string
0 -> Exit the program.
> 2
You Selected 2
Enter String to Copy : Abhinav
The given string copied to array d
Here is the copied string (with pointer): Abhinav
The given string copied to array e
Here is the copied string (without pointer): Abhinav

--------------------------------
1 -> Get Length Of Given String
2 -> Copy string to another array
3 -> Concatonate two strings
4 -> Compare two strings
5 -> Reverse given string
0 -> Exit the program.
> 3
You Selected 3

Enter First string : Hello
```

```
Enter Second string : World

Concatonated string is (with pointer): HelloWorld
Concatonated string is (without pointer): HelloWorld

--------------------------------
1 -> Get Length Of Given String
2 -> Copy string to another array
3 -> Concatonate two strings
4 -> Compare two strings
5 -> Reverse given string
0 -> Exit the program.
> 4
You Selected 4

Enter First string : VIIT

Enter Second string : Algorithm

Second string has more character than the first one.Difference of character is :
5

--------------------------------
1 -> Get Length Of Given String
2 -> Copy string to another array
3 -> Concatonate two strings
4 -> Compare two strings
5 -> Reverse given string
0 -> Exit the program.
> 5
You Selected 5
Enter String to Reverse : VIIT

Reversed string is (with pointer): TIIV
Reversed string is (with pointer): TIIV

--------------------------------
1 -> Get Length Of Given String
2 -> Copy string to another array
3 -> Concatonate two strings
4 -> Compare two strings
5 -> Reverse given string
0 -> Exit the program.
> 0
You Selected 0

Exited!
PS G:\viit notes 2020-21\Sem III SY\Fundamentals Of Data
Structure\Practical\Assignment 1>
```


- **Conclusion:**
  - Used pointers to perform string operations
  - Use of '\0'.