ASSIGNMENT NO: 1

AIM: To perform various string operations using pointer.

1. String length.
2. String compare.
3. String copy.
4. String concate.
5. String reverse.

OBJECTIVE: Performing various string operations such as finding the length of the string, comparing two strings, copying one string into another string, joining two strings or reversing a string can be done by using the concept of pointers.

THEORY: 1) We know that, string is a collection of characters. By writing char x[20] where x is any variable, we declare an array of characters giving it a size of 20.

* 1. A string can also be declared using pointers.
  2. char \*p, this stores the address of x (p=x) ,therefore the value of \*p = x[20].
  3. We can perform different kinds of string functions under pointers in predefined string functions.

SOURCE CODE:

#include<iostream>

using namespace std;

int length(char\*p) /\* p=&str[0] \*/

{

int count = 0;

while (\*p != '\0') {

count++;

p++;

}

return count;

}

void concate(char\*p,char\*q)

{

while(\*p)

{

p++;

}

while(\*q)

{

\*p=\*q;

p++;

q++;

}

\*p='\0';

}

void copy(char \*q, char \*p)

{

while(\*p)

{

\*q = \*p;

p++;

q++;

}

\*q = '\0';

}

void reverse(char\*p)

{

int l, c;

char \*start, \*end, temp;

l = length(p);

start = p;

end = p;

for (c = 0; c < l - 1; c++)

end++;

for (c = 0; c < l/2; c++)

{

temp = \*end;

\*end = \*start;

\*start = temp;

start++;

end--;

}

}

void compare(char\*p,char\*q)

{

while (\*p == \*q) {

if (\*p == '\0' || \*q == '\0')

break;

p++;

q++;

}

if (\*p == '\0' && \*q == '\0')

{

cout<<endl<<"Two string are same...";

}

else

{

cout<<endl<<"Two string are not same...";

}

}

int main()

{

char str1[20],str2[20];

int choice,len;

p:

cout<<"\n\nEnter the process u want to perform on any string : ";

cout<<"\n1.Length\n2.concatenate\n3.copy\n4.reverse\n5.compare\n";

cin>>choice;

switch(choice)

{

case 1: cout<<"\nEnter first string : ";

cin>>str1;

len=length(str1);

cout<<"\nLength of string is : "<<len;

break;

case 2: cout<<"\nEnter first string : ";

cin>>str1;

cout<<"\nEnter second string : ";

cin>>str2;

concate(str1,str2);

cout<<"\nThe result will be :"<<str1;

break;

case 3: cout<<"\nEnter first string : ";

cin>>str1;

cout<<"\nEnter second string : ";

cin>>str2;

copy(str2,str1);

cout<<"\n1st string is copied into 2nd string\nresult of copy : "<<str2;

break;

case 4: cout<<"\nEnter first string : ";

cin>>str1;

reverse(str1);

cout<<"\nThe revrse of the string is :"<<str1;

break;

case 5: cout<<"\nEnter first string : ";

cin>>str1;

cout<<"\nEnter second string : ";

cin>>str2;

compare(str1,str2);

break;

default:cout<<"\nERROR! Invalid Option";

break;

}

char ch;

cout<<"\n\nDo you want to continue (~press Y/N)\n";

cin>>ch;

if(ch=='y'||ch=='Y')

{

goto p;

}

return 0;

}

OUTPUT:

/\*

Enter the process u want to perform on any string :

1.Length

2.concatenate

3.copy

4.reverse

5.compare

1

Enter first string : Nikit

Length of string is : 5

Do you want to continue (~press Y/N)

y

Enter the process u want to perform on any string :

1.Length

2.concatenate

3.copy

4.reverse

5.compare

4

Enter first string : nikit

The revrse of the string is :tikin

Do you want to continue (~press Y/N)

n

--------------------------------

Process exited after 42.41 seconds with return value 0

Press any key to continue . . .

\*/

CONCLUSION : 1) It is a way of using a contagious chunk of memory.

2) Pointers can hold only the address of the string and not the characters of the strings 3)Pointers are basically used to store and manage address dynamically.

1. Pointers increase the processing speed.
2. Pointers save the memory.