	PUNE INSTITUTE OF COMPUTER TECHNOLOGY PUNE - 411043	
	Department of Electronics & Telecommunication	
	ASSESMENT YEAR: 2021-2022	CLASS: SE-5
	SUBJECT: DATA STRUCTURES	
EXPT No:	LAB Ref: SE/2021-22/	Starting date: 15\11\2021
	Roll No:22108	Submission date: 20\11\2021
Title:	**String Operations (With Pointers)**	
Problem statement	Perform following string operations without using the library functions A. With pointers to arrays b. Without pointers to arrays 1. Substring, 2. Palindrome, 3. Compares, 4. Copy, 5. Reverse	
Prerequisites:	Basics of C programming	
	Decision making and loop controls	
	Choice based program	
	Strings	
Objectives:	Learn to create and display a string.	
	Implement various operation on array to understand its effect on data.	
	Verify operation with and without pointer	
Theory:		

•

Strings –

1. String is not a datatype in C Language, but it is considered as a data structure of characters stored in an array.
2. Every String in C end with '\0' or Null.
3. Each element of string array, which are letters, occupy 1 b1t of memory.

Declaring a String:

char str[size]; or char

str[size] = "String"

Functions -

- It is a self-contained block of statements that perform task of some kind.
- C program may have one or more functions
- C program must have at least one function i.e. main()
- There is no limit on number of functions
- Each function is called in a sequence specified by the function calls in main()

After each function has done its job, control returns to next location from where it has been called.

Pointers –

- The pointer in C language is a variable which stores the address of another variable. This variable can be of type int, char, array, function, or any other pointer. The size of the pointer depends on the architecture. However, in 32-bit architecture the size of a pointer is 2 byte.
- The pointer in c language can be declared using * (asterisk symbol). It is also known as indirection pointer used to dereference a pointer.
- The general form of a pointer variable declaration is − type *var-name;

•

Algorithm	1) Algorithm for create an array • Start • Declare a string array char str [10] with maximum size 10. • Ask user to enter how elements he wants in an array save size in n. • For(int i = 0;i<n;i++) • Scan element one by one by using "%s" • End of for loop • end 2) Algorithm to display an array • Start • Declare a string array char str [10] with maximum size 10. • Ask user to enter how elements he wants in an array save size in n. • For(int i = 0;i<n;i++) • Scan element one by one by using "%s" • End of for loop • Print the string using 'printf("%s", str); • End 3) Algorithm to display substring • Start • Declare a string array char str [10] with maximum size 10. • Ask user to enter how elements he wants in an array save size in n. • For(int i = 0;i<n;i++) • Scan element one by one by using "%s" • End of for loop • Print the string using 'printf("%s", str); • Declare variables for size and beginning position of substring (substr_size, position) and a iterating variable c = 0 • Ask user the position and size of substring • While(c<substr_size) • Substr[c] = str[position+c-1]; Here the substring array is copying the char elements from the main string from the user defined position till the size of substring is fulfilled.

•

Increment c by 1
- End of while loop
- Substr[c] = '\0' assigning the last position of substring as Null to end the string.
- Print the substring.
- End

4) Algorithm to check for palindrome.
- Start
- Declare a string array char str [10] with maximum size 10.
- Ask user to enter how elements he wants in an array save size in n.
- For(int i = 0;i<n;i++)
- Scan element one by one by using "%s"
- End of for loop
- Print the string using 'printf("%s", str);
- Declare variable for checking if palindrome is there (flag = 0)
- For(i = 0; i>n; i++)
- If(str[i]!=str[size-i-1]) flag = 1 and break from the loop
- End of for loop
- Check if flag is 1, if true then string doesn't have palindrome, else vice versa.
- End

5) Algorithm to compare two strings
- Start
- Declare a string array char str [10] with maximum size 10.
- Ask user to enter how elements he wants in an array save size in n.
- For(int i = 0;i<n;i++)
- Scan element one by one by using "%s"
- End of for loop
- Print the string using 'printf("%s", str);
- Declare variables for size of new string (new_size), compare checker to check if the two strings are same (cmp = 1), and finally character array for new string to be compared char cmprstr[20];
- Ask the user for the size of new string
- For(int i = 0;i<new_size;i++)
- Scan element one by one by using "%s" for cmprstr string
- End of for loop
- Cmprstr[new_size] = '\0'
- i = 0
- While(str[i] != '\0' && cmprstr[i] != '\0')

	• • If(str[i] != cmprstr[i]) • Cmp = 0 and break from the loop • Increment i • End of while loop

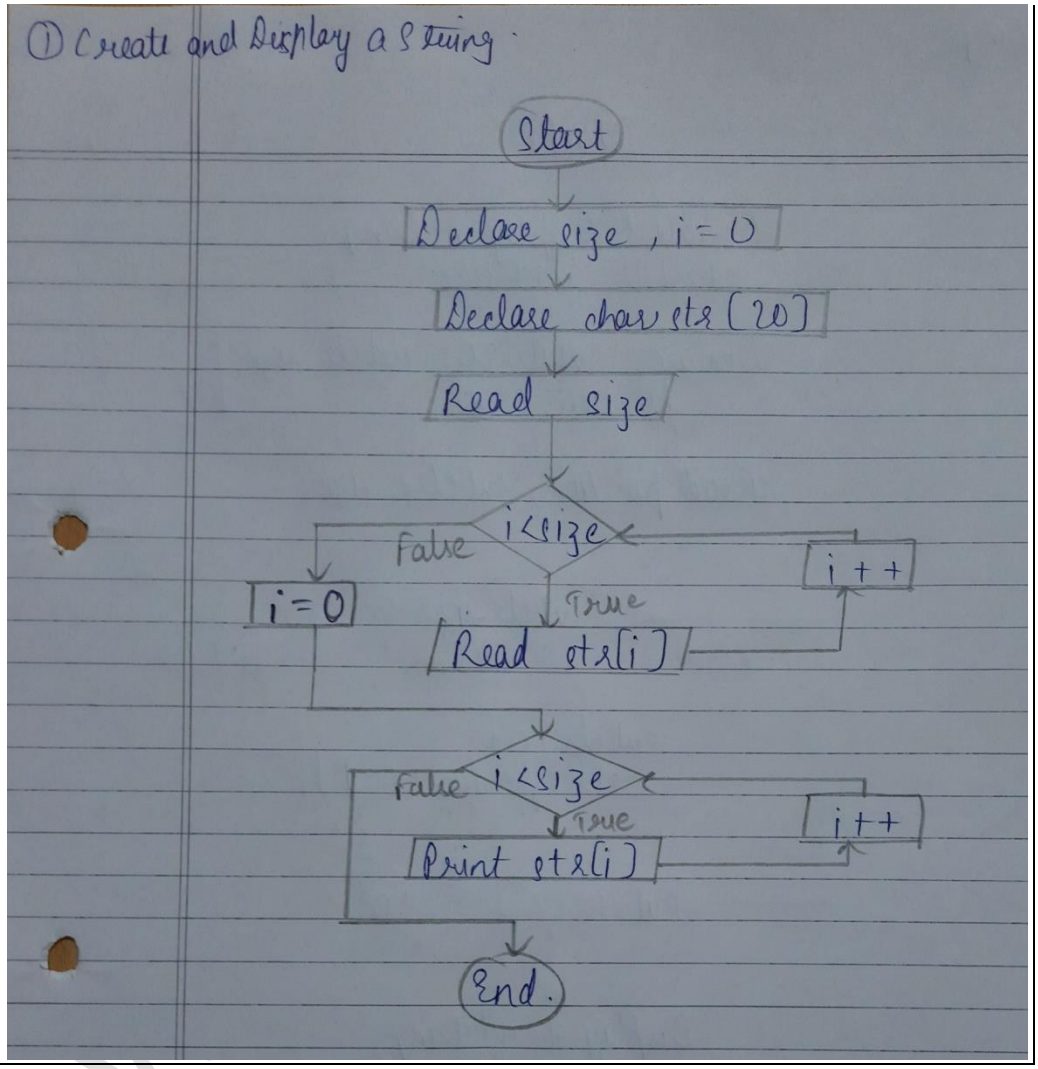
•

If cmp == 0, strings are not the same and print the result
- If cmp == 1, strings are the same and print the result.
- End

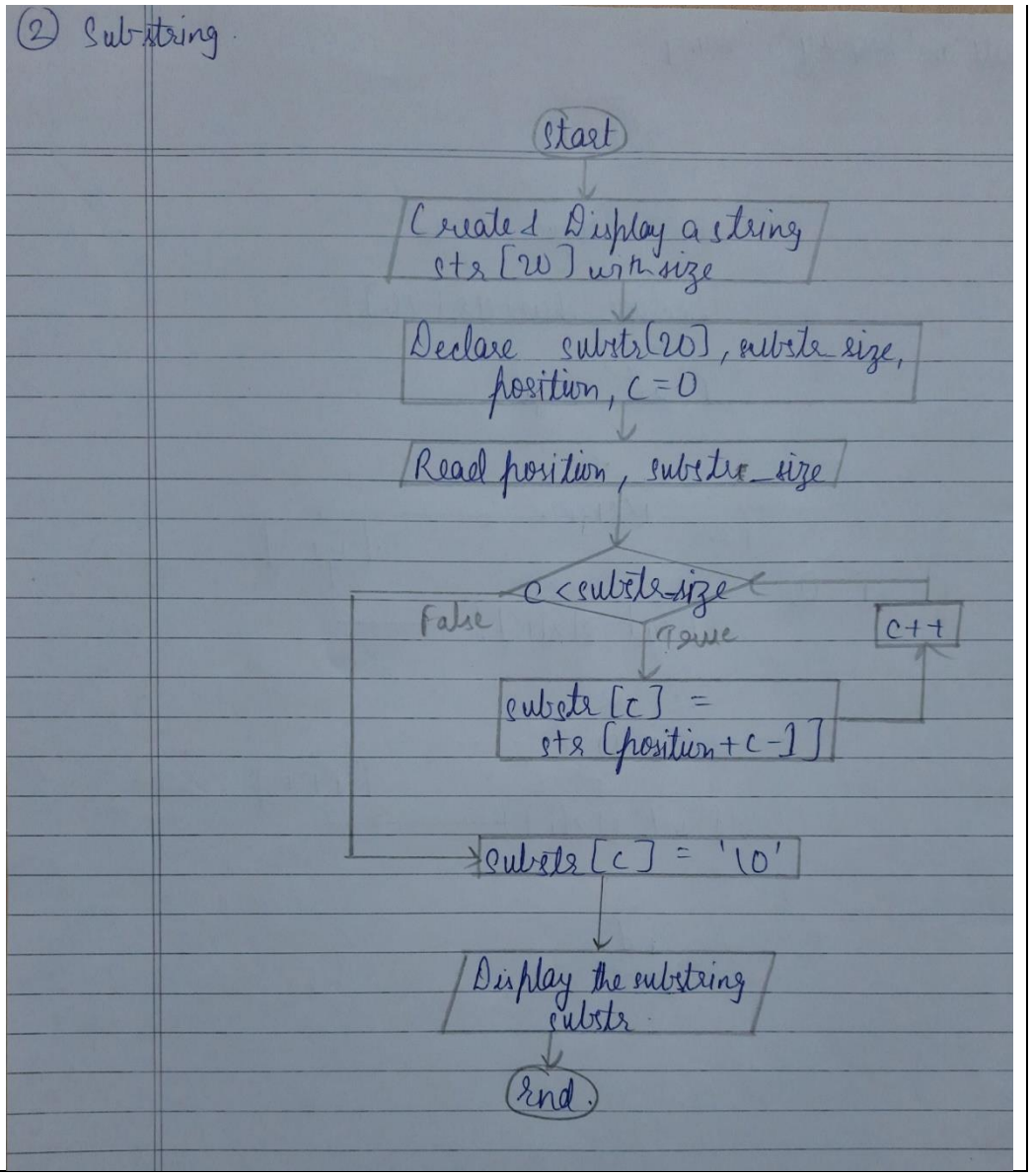
6) Algorithm for copying a string
- Start
- Declare a string array char str [10] with maximum size 10.
- Ask user to enter how elements he wants in an array save size in n.
- For(int i = 0;i<n;i++)
- Scan element one by one by using "%s"
- End of for loop
- Print the string using 'printf("%s", str);
- Declare another character array for string as char str1[20], will have the same size n as the main string str
- For(i = 0; i<n; i++)
- Str1[i] = str[i] assigning the elements of main string to the new string variable created
- End of for loop
- Str1[n] = '\0'
- Print both, the main string str and the newly created string str1.
- End

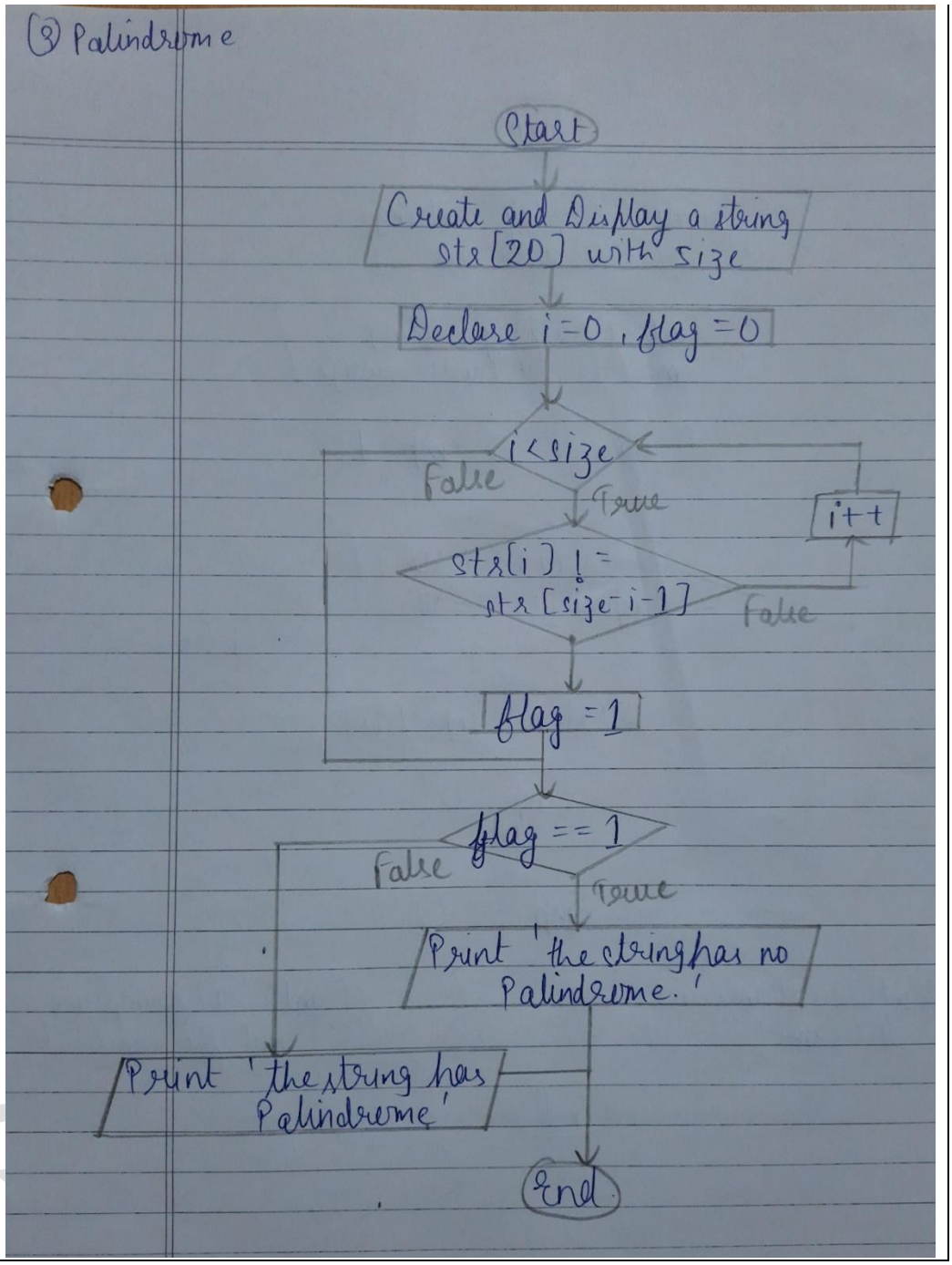
7) Algorithm for Reversing a String
- Start
- Declare a string array char str [10] with maximum size 10.
- Ask user to enter how elements he wants in an array save size in n.
- For(int i = 0;i<n;i++)
- Scan element one by one by using "%s"
- End of for loop
- Print the string using 'printf("%s", str);
- Declare a new string char revstr[20], will have the same size n as the main string str
- For(i = 0; i< n; i++)
- Revstr[i] = str[n-i-1]
- End of for loop
- Revstr[n] = '\0'
- Print the string revstr
- End

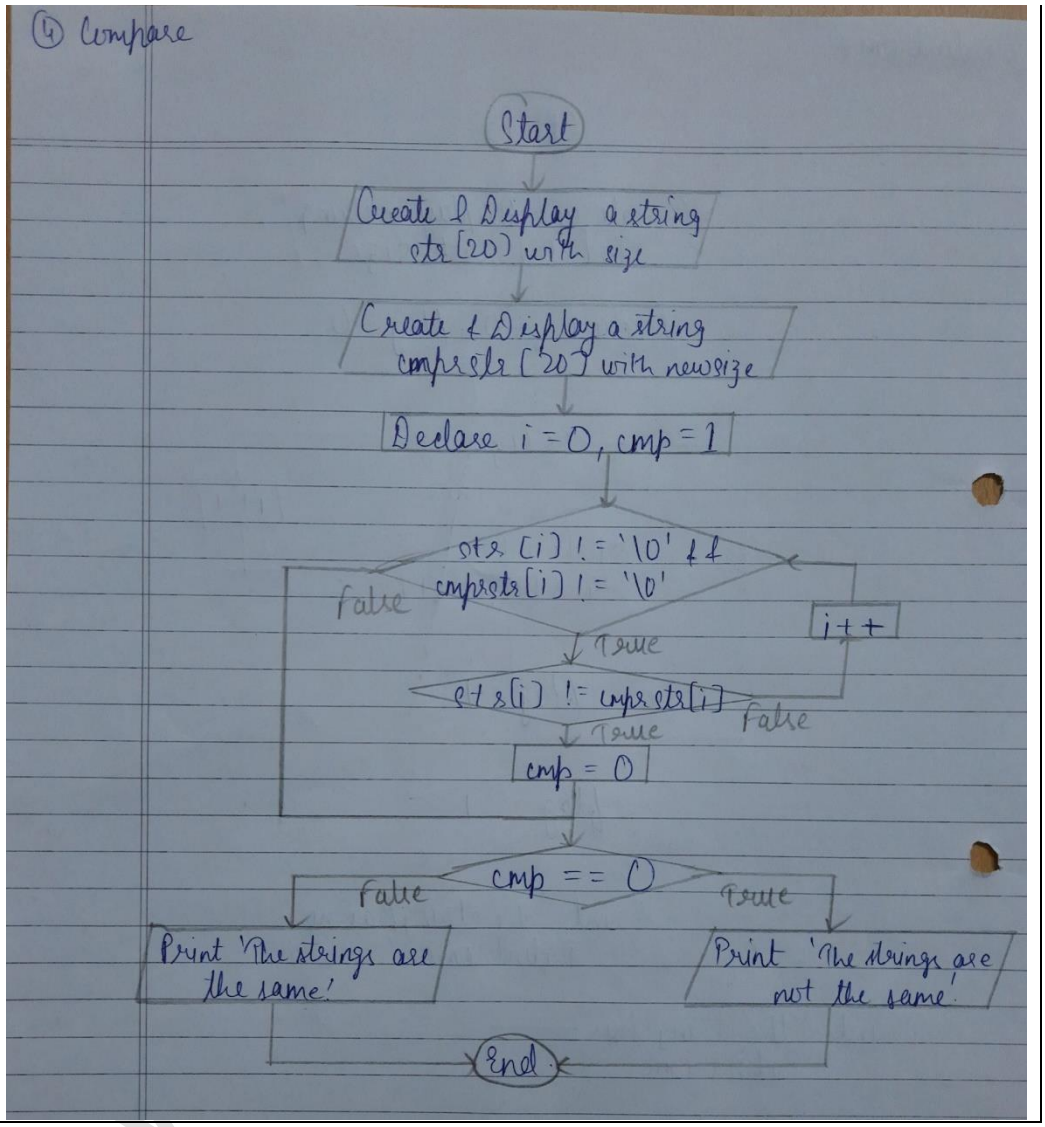
Flow-chart

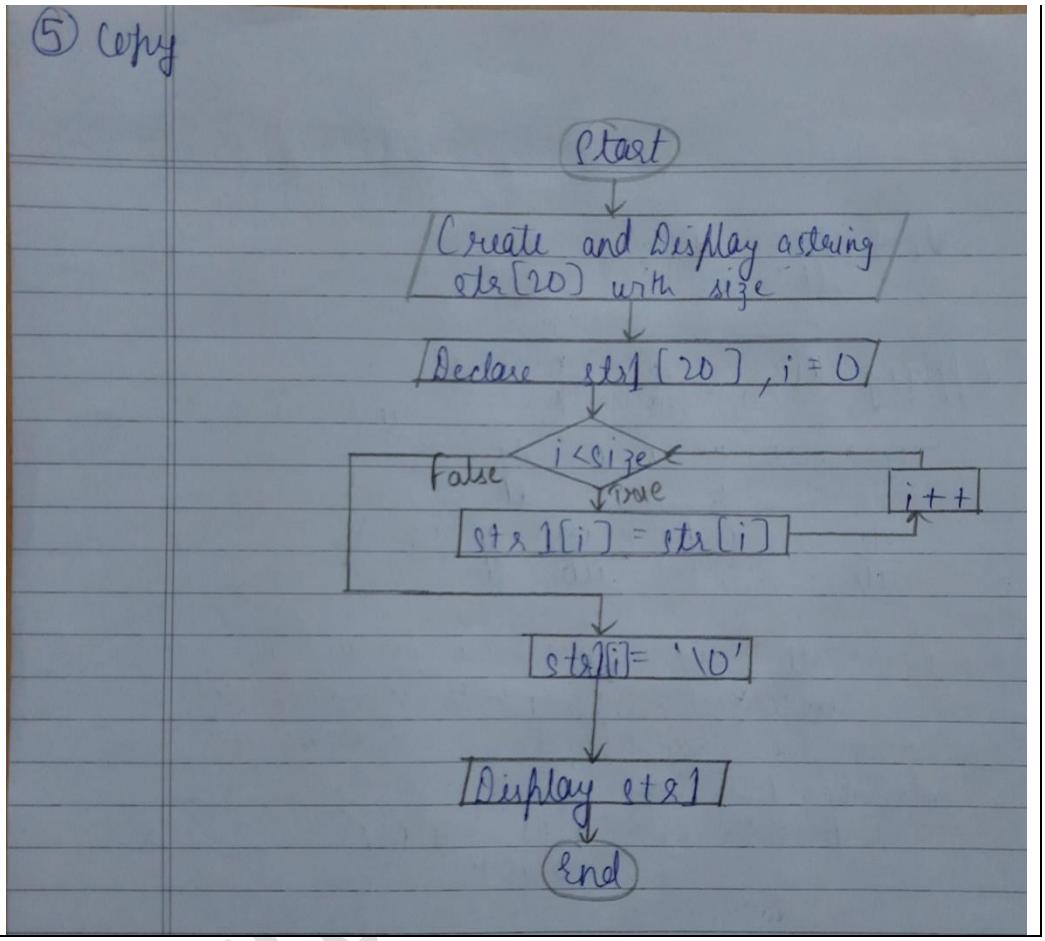

① Create and Display a String.
Start
Declare size, i = 0
Declare char str[20]
Read size
i<size
False
True
i = 0
Read str[i]
i++
i<size
False
True
Print str[i]
i++
End.

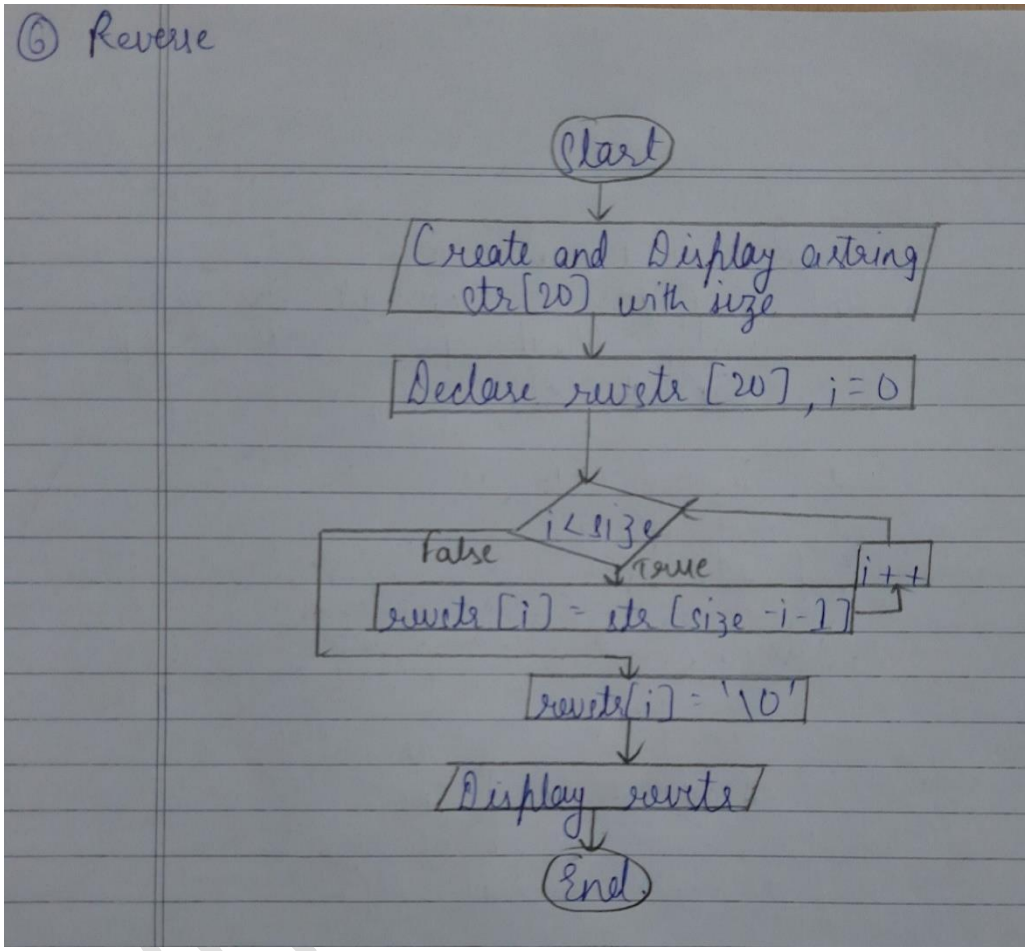
② Substring.


Start
Create & Display a string str[20] with size
Declare substr[20], substr_size, position, c=0
Read position, substr_size
c < substr_size
False
True
c++
substr[c] = str[position+c-1]
substr[c] = '\0'
Display the substring substr
End.


③ Palindrome
Start
Create and Display a string str[20] with size
Declare i=0, flag=0
i<size
False
True
i++
str[i] != str[size-i-1]
False
flag = 1
flag == 1
False
True
Print 'the string has no Palindrome.'
Print 'the string has Palindrome'
End


④ Compare
Start
Create & Display a string str[20] with size
Create & Display a string cmpstr[20] with newsize
Declare i=0, cmp=1
str[i] != '\0' && cmpstr[i] != '\0'
False
True
i++
str[i] != cmpstr[i]
True
False
cmp = 0
cmp == 0
False
True
Print 'The strings are the same'
Print 'The strings are not the same'
End


⑤ Copy
Start
Create and Display a string str[20] with size
Declare str1[20], i = 0
i<size
False
True
i++
str1[i] = str[i]
str[i] = '\0'
Display str1
End

	
ERROR and REMEDY	-

Code	#include <stdio.h>
	#include <stdbool.h>
	
	void substr(int size, char *ptr)
	{
	int substr_size, position, c =
	0; char substr[20]; char
	*sub_ptr = substr;
	printf("\n");
	printf("Enter the position of substring: ");
	scanf("%d", &position);
	printf("\n"); if
	(position > size)
	{
	printf("The position of substring beginning cannot be greater than the size of main string! Try Again...."); return;
	}
	printf("Enter the size of substring: ");
	scanf("%d", &substr_size);
	if (substr_size > (size-position))
	{ printf("The size of substring apperently is exceeding the string itself if counted from %d position\nSubstring Size = %d\nRemaining no of letters in string from %d position = %d\nThis is why, Substring cannot be extracted! Try Again....", position, substr_size, position, (size-position));
	return;
	}
	
	while (c < substr_size)
	{
	*(sub_ptr+c) = *(ptr+position + c - 1);
	c++;
	}
	*(sub_ptr+c) = '\0';
	
	printf("\n"); printf("Required
	Substring is: '");
	for (int i = 0; i < substr_size; i++)
	{
	printf(" %c", *(sub_ptr+i));
	}
	printf("'");

```
}

void palindrome_check(int size, char *ptr)
{
   int i, flag = 0;     for (i
= 0; i < size; i++)
   {
      if (*(ptr+i) != *(ptr+size - i - 1))
      {
flag = 1;
break;
      }
}
   if (flag == 1)
   {
      printf("\nThe string didn't have palindrome!");
   }
else
   {
      printf("\nThe String has Palindrome!!");
   }
}

void comparestr_check(int size, char *ptr)
{ int newsize, i = 0,
  cmp; char cmprstr[20];
  char *cmpr_ptr =
  cmprstr;

  printf("\nEnter the size of new string to be compared: ");
  scanf("%d", &newsize);

   printf("\nEnter the characters of the string one by one:\n");
for (int i = 0; i < newsize; i++)
   {
      scanf(" %c", (cmpr_ptr+i));
   }
```

```
   printf("The main string: ");
   for (i = 0; i<size; i++)
   {
      printf(" %c", *(ptr+i));
   }
printf("\n");

   printf("The new string to be compared to the main string: ");
printf(" %c", *(cmpr_ptr+i));
   printf("\n");

   i = 0;
   while (*(ptr+i) != '\0' && *(cmpr_ptr+i) != '\0')
   {
cmp = 1;
      if (*(ptr+i) != *(cmpr_ptr+i))
      {
         cmp = 0; //Compare is False

break;
}        i++;
   }
   if (cmp == 1)
   {
      printf("\nThe strings are same!");
   }
   else if (cmp == 0)
   {
      printf("\nThe strings are not the same");
   }
}

void copystr(int size, char *ptr)
{     char str1[20];
char *ptr1 = str1;
  int i;
  //Here size of both strings will be same as we are just copying them.
  printf("Here we will take one more variable as str1 to copy string from
  str");
```

```
  for (i = 0; i < size; i++)
  {
      *(ptr1+i) = *(ptr+i);
   }     str1[i] = '\0';
printf("\n");     printf("String in
str variable: ");     for ( i = 0; i <
size; i++)
   {
      printf(" %c", *(ptr+i));
   }
   printf("\n");

   printf("String in str1 variable: ");
for ( i = 0; i < size; i++)
   {
      printf(" %c", *(ptr+i));
   }
   printf("\n");
}

void reversestr(int size, char *ptr)
{     char
revstr[20];
   char *rev_ptr = revstr;
   int i;
   for (i = 0; i < size; i++)
   {
      *(rev_ptr+i) = *(ptr+size - i - 1);
   }
   *(rev_ptr+i) = '\0';
   printf("\nThe reverse of the string is: ");
   for ( i = 0; i < size; i++)
   {
      printf(" %c", *(rev_ptr+i));
   }
}
```

```
void end()
{
   printf("\n++++++++++This is the end of Execution!++++++++++");
}

int main()
{ int size, i,
  choice; char
  str[20]; char *ptr
  = str;

  printf("+++++++++++++++String Operations!+++++++++++++++");
    printf("First enter the size and the string itself for operating on it, this is the
main string!\n");

   printf("Enter the size of string: ");
   scanf("%d", &size);
   printf("Enter characters one by one:\n");
   for (i = 0; i < size; i++)
   {
      scanf(" %c", (ptr+i));
   }

   printf("The string entered: ");
for ( i = 0; i < size; i++)
   {
      printf(" %c", *(ptr+i));
   }
   printf("\n");

   printf("\nThere are 5 operations on string, and you need to choose to perform
one of them by number!\nThe choices are:\n1) Substring\n2) Check for
Palindrome\n3) Compare two strings\n4) Copy\n5) Reverse a string\n6)
Exit\n");
   scanf("%d", &choice);

   switch (choice)
   {
   case 1: //Substring Operation
```

```
   {
printf("\n");
      printf("You have chosen Substring Operation!\n");
substr(size, ptr);        end();        break;
   }
   case 2: //Palindrome Operation
   {
printf("\n");
      printf("You have chosen Palindrome Check Operation!\n");
      palindrome_check(size, str);
      end();
break;
   }
   case 3: //Compare Operation
   {
printf("\n");
```

```
        printf("You have chosen Compare Operation!\n");
comparestr_check(size, ptr);        end();        break;
    }
    case 4: //Copy Operation
    {
        printf("\n");
        printf("You have chosen Copy Operation!\n");
copystr(size, ptr);
        end();
break;
    }
    case 5: //Reverse Operation
    {
        printf("\n");
        printf("You have chosen Reverse Operation!\n");
reversestr(size, ptr);
        end();
break;
    }
case 6:
    {
        printf("\n");
end();
break;
    }
default:
        printf("\nEnter valid choice, number between 1 to 6");
break;
    }

    return 0;
}
```

Output	

1) Substring –

Successful:

++++++++++++++++String Operations!++++++++++++++++
First enter the size and the string itself for operating on it, this is the main string!
Enter the size of string: 4 Enter characters one by one: f
o u
r
The string entered: four

There are 5 operations on string, and you need to choose to perform one of them by number!

The choices are:
1) Substring
2) Check for Palindrome
3) Compare two strings
4) Copy
5) Reverse a string
6) Exit
1

You have chosen Substring Operation!

Enter the position of substring: 2

Enter the size of substring: 3

Required Substring is 'our'.
++++++++++This is the end of Execution!++++++++++

Unsuccessful:

++++++++++++++++String Operations!++++++++++++++++
First enter the size and the string itself for operating on it, this is the main string!
Enter the size of string: 4 Enter characters one by one:

f
o
u
r
The string entered: four

There are 5 operations on string, and you need to choose to perform one of them by number! The choices are:

1) Substring
2) Check for Palindrome
3) Compare two strings
4) Copy
5) Reverse a string
6) Exit

1

You have chosen Substring Operation!

Enter the position of substring: 3

Enter the size of substring: 3
The size of substring apperently is exceeding the string itself if counted from 3 position Substring Size = 3
Remaining no of letters in string from 3 position = 1

This is why, Substring cannot be extracted! Try Again....
++++++++++This is the end of Execution!++++++++++

2) Palindrome –
Successful:
+++++++++++++++String Operations!+++++++++++++++
First enter the size and the string itself for operating on it, this is the main string!
Enter the size of string: 9 Enter characters one by one: m
a
l
a
y
a
l
a
m
The string entered: malayalam

There are 5 operations on string, and you need to choose to perform one of them by number! The choices are:
1) Substring
2) Check for Palindrome
3) Compare two strings
4) Copy
5) Reverse a string
6) Exit
2

You have chosen Palindrome Check Operation!

The String has Palindrome!!
++++++++++This is the end of Execution!++++++++++

Unsuccessful:
+++++++++++++++String Operations!+++++++++++++++
First enter the size and the string itself for operating on it, this is the main string!
Enter the size of string: 5 Enter characters one by one: t
h
r
e
e
The string entered: three

There are 5 operations on string, and you need to choose to perform one of them by number! The choices are:

1) Substring
2) Check for Palindrome
3) Compare two strings
4) Copy
5) Reverse a string
6) Exit

2

You have chosen Palindrome Check Operation!

The string didn't have palindrome!
++++++++++This is the end of Execution!++++++++++

3) Compare –
Successful:
++++++++++++++++String Operations!++++++++++++++++
First enter the size and the string itself for operating on it, this is the main string!
Enter the size of string: 4 Enter characters one by one:
f
o
u
r
The string entered: four

There are 5 operations on string, and you need to choose to perform one of them by number! The choices are:

1) Substring
2) Check for Palindrome
3) Compare two strings
4) Copy
5) Reverse a string
6) Exit

3

You have chosen Compare Operation!

Enter the size of new string to be compared: 4

Enter the characters of the string one by
one: f o u
r
The main string: four

The new string to be compared to the main string: four

The strings are same!
++++++++++This is the end of Execution!++++++++++

Unsuccessful:
+++++++++++++++String Operations!+++++++++++++++
First enter the size and the string itself for operating on it, this is the main string! Enter the size of string: 3 Enter characters one by one: o n e
The string entered: one

There are 5 operations on string, and you need to choose to perform one of them by number! The choices are:
1) Substring
2) Check for Palindrome
3) Compare two strings
4) Copy
5) Reverse a string
6) Exit
3

You have chosen Compare Operation!

Enter the size of new string to be compared: 3

Enter the characters of the string one by
one: s i
x
The main string: one
The new string to be compared to the main string: six

The strings are not the same
++++++++++This is the end of Execution!++++++++++

4) Copy –
+++++++++++++++String Operations!+++++++++++++++
First enter the size and the string itself for operating on it, this is the main string!
Enter the size of string: 5 Enter characters one by one: n
i
g
h
t

The string entered: night

There are 5 operations on string, and you need to choose to perform one of them by number! The choices are:

1) Substring
2) Check for Palindrome
3) Compare two strings
4) Copy
5) Reverse a string
6) Exit

4

You have chosen Copy Operation!
Here we will take one more variable as str1 to copy string from str
String in str variable: night
String in str1 variable: night

++++++++++This is the end of Execution!++++++++++

5) Reverse –

++++++++++++++++String Operations!++++++++++++++++
First enter the size and the string itself for operating on it, this is the main string! Enter the size of string: 6 Enter characters one by one: g a r r i
x
The string entered: garrix

There are 5 operations on string, and you need to choose to perform one of them by number! The choices are:

1) Substring
2) Check for Palindrome
3) Compare two strings
4) Copy
5) Reverse a string
6) Exit

5

You have chosen Reverse Operation!

The reverse of the string is: xirrag

	++++++++++This is the end of Execution!++++++++++
CONCLUSION:	
	In this practical, we learn about various concepts related to string. We also successfully implemented a C program for performing various string operations like substring, palindrome, compare, copy, reverse, without using library functions. Here, we also used pointers.
REFERENCES:	
	Seymour Lipschutz, Data Structure with C, Schaum's Outlines, Tata McGrawHill
	E Balgurusamy - Programming in ANSI C, Tata McGraw-Hill (Third Edition)
	Yashavant Kanetkar- Let Us C, BPB Publication, 8th Edition.

Continuous Assessment for DS AY 2021-22			
RPP (5)	**SPO (5)**	**Total (10)**	**Signature:**
			Assessed By: Mr. V. B. Vaijapurkar
Start date	**Submission date**		**Date:**
15/11/2021	**20/11/2021**		**Roll. No.22108**
***Regularity, Punctuality, performance** ***Submission, Presentation, orals**			