STRUCTURES-STRINGS-POINTERS

BY ASRITHA MEKA, AP1911001022,CSE-G

25)

OBJECTIVE:

Write a c program using structure for entering details of the five students as name, admission number, date of birth, department and display all the details.

CODE:

```
#include <stdio.h>
typedef struct Date{
 int month;
 int day;
 int year;
}date;
typedef struct Stud{
 char name[40];
 int adminNo;
 date DOB;
 char dept[40];
}stud;
int main()
{
 stud s[8];
 int i;
 printf("Enter details: \n");
 for(i=0;i<5;i++){
  printf("Enter\n");
  printf("Name of stud");
  scanf("%s",s[i].name);
  printf("Admin No of stud: ");
  scanf("%d",&s[i].adminNo);
  printf("Date of Birth of stud:\n");
  printf("Date: ");
  scanf("%d",&s[i].DOB.day);
  printf("Month: ");
  scanf("%d",&s[i].DOB.month);
  printf("year: ");
  scanf("%d",&s[i].DOB.year);
  printf("Department of stud: ");
  scanf("%s",s[i].dept);
 }
 printf("\n Details: \n");
 for(i=0;i<5;i++){
  printf("Name of stud: %s\n",s[i].name );
  printf("Admin No of stud: %d\n",s[i].adminNo );
  printf("Date of Birth of stud: %d/%d/%d\n",s[i].DOB.day,s[i].DOB.month,s[i].DOB.year);
  printf("Department of stud: %s\n\n",s[i].dept);
 }
 return 0;
}
```

OUTPUT:

```
Enter details:
Enter
Name of stud asritha
Admin No of stud: 224
Date of Birth of stud:
Date: 090801
Month: september
year: Department of stud: Enter
Name of stud^C
```

26)

Objective

Write a c program to find the length of string using pointers

Code:

```
#include <stdio.h>
#include <string.h>
int main()
{
  char x[500];
  int len;
  printf("Enter a string for calculating len\n");
  gets(x);
  len = strlen(x);
  printf("Len of string = %d\n", len);
  return 0;
}
```

Output:

```
Enter a string for calculating len
as  1234
Len of string = 4
```

27)

Objective:

Write a c program to copy one string to another using pointers

CODE:

```
#include <stdio.h>
void copystr(char *dest,char *src)
{
            while(*src!='\0')
            *dest++=*src++;
            *dest='\0';
}
int main(int argc, char const *argv[])
{
 char string1[70],string2[70];
 printf("Enter the string:\n");
 gets(string1);
 copystr(string2,string1);
 printf("String 1:\n");
 printf("%s\n",string1 );
 printf("string2:\n");
 printf("%s\n",string2 );
 return 0;
}
```

Output:

```
Enter the string:
25
String 1:
25
string2:
25
```

28)

Objective:

Write a c program to compare two strings using pointers

Code:

```
#include<stdio.h>

int compare_string(char*, char*);

main()
{
	char first[400], second[400], result;

	printf("Enter the 1st string\n");
	gets(first);

	printf("Enter the 2nd string\n");
	gets(second);

	result = compare_string(first, second);

	if ( result == 0 )
	printf("Both strings are same.\n");
	else
	printf("Entered strings are not equal.\n");

	return 0;
}

int compare_string(char *first, char *second)
```

```
{
  while(*first==*second)
  {
        if ( *first == '\0' || *second == '\0' )
        break;

        first++;
        second++;
  }
  if( *first == '\0' && *second == '\0' )
        return 0;
  else
        return -1;
}
```

Output:

```
Enter the 1st string
76
Enter the 2nd string
99 kk
Entered strings are not equal.
```

29)

Objective:

Write a c program to find the reverse of a string recursively and non-recursively.

Code:

```
#include <stdio.h>
void recRev(char *a, int star, int end)
{
  char temp;

  if (star >= end)
        return;

  temp = *(a+star);
  *(a+star) = *(a+end);
```

```
   *(a+end)   = temp;

   recRev(a, ++star, --end);
}
int string_ln(char* p){
 int count = 0;
 while(*p != '\0'){
  count++;
  p++;
 }
 return count;
}
void rev(char *str)
{
   int len, i;
   char *star, *end, temp;

   len = string_ln(str);

   star = str;
   end = str;

   for (i = 0;i < (len - 1 ) ; i++ )
            end++;
   for ( i = 0 ; i < len/2 ; i++ )
   {
            temp = *end;
            *end = *star;
            *star = temp;

            star++;
            end--;
   }
}
int main(int argc, char const *argv[])
{
 char str1[70];
 printf("Enter the string:\n");
 gets(str1);
 printf("reversing the string by using recursion\n");
 recRev(str1,0,string_ln(str1)-1);
 printf("%s\n",str1);
 printf("Reversing the reversed string by using iteration\n");
 rev(str1);
 printf("%s\n",str1);
 return 0;
}
```

Output:

```
Enter the string:
67
reversing the string by using recursion
76
Reversing the reversed string by using iteration
67
```