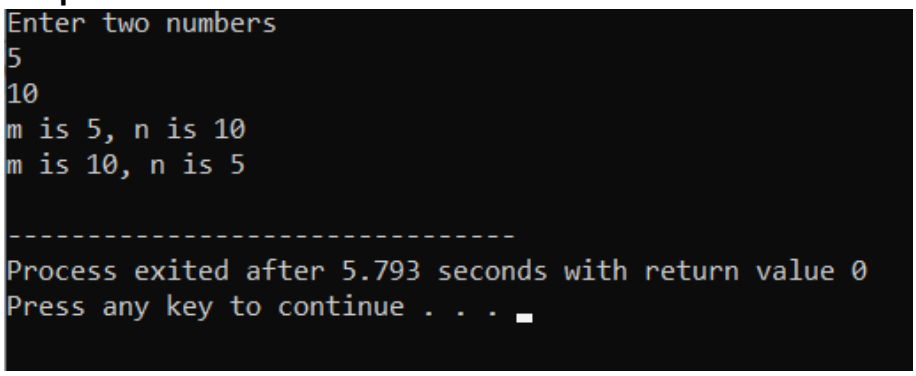
ASSIGNMENT-01

1. WAP to swap the values of two variables using a user defined function.(call by reference)

Ans:

```
 #include<stdio.h>
void swap (int *a, int *b) {
  int temp;
  temp = *a;
  *a = *b;
  *b = temp;
}
int main() {
    int m,n;
    printf("Enter two numbers\n");
    scanf("%d %d",&m,&n);
    printf("m is %d, n is %d\n", m, n);
    swap(&m, &n);
    printf("m is %d, n is %d\n", m, n);
    return 0;
}
```

Output:



2.Write a user defined function to Sort an array of integer with pointer as argument.

Ans:

```
#include <stdio.h>
void sort(int n, int* ptr)
{
    int i, j, t;
    for (i = 0; i < n; i++) {

        for (j = i + 1; j < n; j++) {

            if (*(ptr + j) < *(ptr + i)) {
```

```
            t = *(ptr + i);
            *(ptr + i) = *(ptr + j);
            *(ptr + j) = t;
         }
      }
   }
   for (i = 0; i < n; i++)
      printf("%d ", *(ptr + i));
}
int main()
{
   int n;;
   int arr[20];
        int i;
        printf("Enter the limit");
        scanf("%d",&n);
        printf("Enter the array");
        for(i=0;i<n;i++)
        {
                scanf("%d",&arr[i]);
        }
   sort(n, arr);
   return 0;
}
```

Output:

```
Enter the limit5
Enter the array2
1
9
4
3
1 2 3 4 9
--------------------------------
Process exited after 6.8 seconds with return value 0
Press any key to continue . . .
```

3.Write a user defined function to search an element in an integer array with pointer as argument.

Ans:

```
#include <stdio.h>
int binarySearch(int* ptr2, int l, int r, int x)
```

```
{
   if (r >= l)
        {
      int mid = l + (r - l) / 2;
      if (*(ptr2+mid) == x)
      return mid;
      if (*(ptr2+mid) > x)
        return binarySearch(ptr2, l, mid - 1, x);
      return binarySearch(ptr2, mid + 1, r, x);
        }
        return -1;
}
void search(int n, int* ptr)
{
   int i, j, t,x;
   for (i = 0; i < n; i++) {

      for (j = i + 1; j < n; j++) {

         if (*(ptr + j) < *(ptr + i)) {

            t = *(ptr + i);
            *(ptr + i) = *(ptr + j);
            *(ptr + j) = t;
         }
      }
   }
   printf("Enter element to search\n");
   scanf("%d",&x);
        int result = binarySearch(ptr, 0, n - 1, x);
   (result == -1) ? printf("Element is not present in array")
                  : printf("Element is present at index %d", result);
}
int main()
{
   int n;;
   int arr[20];
        int i;
        printf("Enter the limit");
        scanf("%d",&n);
        printf("Enter the array");
        for(i=0;i<n;i++)
        {
                  scanf("%d",&arr[i]);
        }
   search(n, arr);
   return 0;
```

}

Output:

```
Enter the limit5
Enter the array2
1
9
3
4
Enter element to search
5
Element is not present in array
--------------------------------
Process exited after 9.619 seconds with return value 0
Press any key to continue . . .
```

4.WAP to print a string in reverse order using user defined function having pointer as an argument.

Ans:

```
#include <stdio.h>
#include <string.h>
void reverseString(char* str)
{
    int l, i;
    char *begin_ptr, *end_ptr, ch;
    l = strlen(str);
    begin_ptr = str;
    end_ptr = str;
    for (i = 0; i < l - 1; i++)
        end_ptr++;
    for (i = 0; i < l / 2; i++) {
        ch = *end_ptr;
        *end_ptr = *begin_ptr;
        *begin_ptr = ch;
        begin_ptr++;
        end_ptr--;
    }
}
int main()
{
        int i;
    char str[100];
    printf("Enter a string:\n");
    scanf("%s",str);
    printf("Enter a string: %s\n", str);
    reverseString(str);
    printf("Reverse of the string: %s\n", str);
```

```
    return 0;
}
```

Output:

```
Enter a string:
trideep
Enter a string: trideep
Reverse of the string: peedirt

--------------------------------
Process exited after 2.687 seconds with return value 0
Press any key to continue . . .
```

5.WAP to sort some strings in dictionary order using user pre-defined function having pointer as an argument.

Ans:

```
#include<stdio.h>
#include<string.h>
void order(char str[][50],int n)
{
        int i,j;
        char temp[50];
  for(i=0;i<n-1;i++)
  {
   for(j=i+1;j<n;j++)
   {
    if(strcmp(str[i],str[j])>0)
    {
     strcpy(temp,str[i]);
     strcpy(str[i],str[j]);
     strcpy(str[j],temp);
    }
   }
  }
  printf("\nIn lexicographical order: \n");
  for(i=0;i<n;i++)
  puts(str[i]);
}
void main()
{
        int n;
        int i;
        int m=50;
  char str[n][m];
  printf("Enter number of words\n");
```

```
   scanf("%d",&n);
   printf("Enter %d Words:\n",n);
   for(i=0;i<n;i++)
   scanf("%s[^\n]",str[i]);
   order(str,n);
}
```

Output:

```
Enter number of words
4
Enter 4 Words:
appke
berry
ant
cow

In lexicographical order:
ant
appke
berry
cow

--------------------------------
Process exited after 11.22 seconds with return value 4
Press any key to continue . . .
```

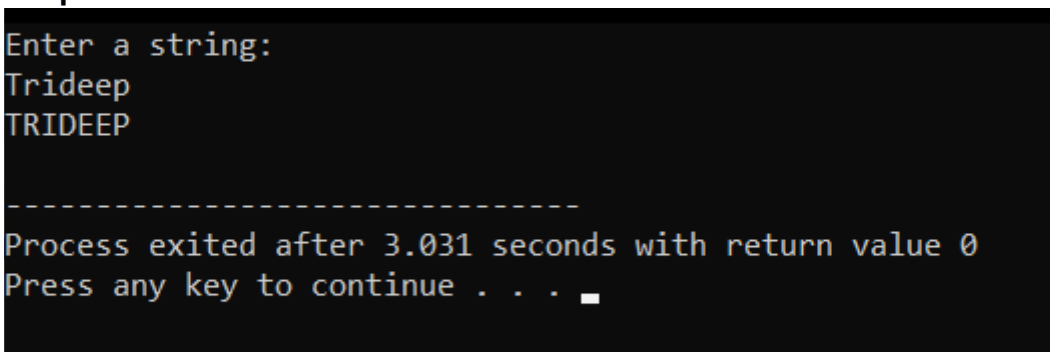
---

6. WAP to convert given string into uppercase if it is in lowercase using user pre-defined function having pointer as an argument.

Ans:

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
	char str[50];
	int i = 0;
	printf("Enter a string:\n");
	fgets(str,50,stdin);
	while(*(str+i)!='\0')
	{
		if(*(str+i)>=97 && *(str+i)<=122)
		{
			*(str+i)=*(str+i)-32;
		}
		i++;
	}
	printf("%s",str);
	return 0;
}
```

Output:



7.WAP to check given string is substring of another given string without using library function & also count how many times the given sub string is present.

Ans:

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

char str[100], sub[100];
int count = 0, count1 = 0;

void main()
{
   int i, j, l, l1, l2;

   printf("\nEnter a string : ");
   scanf("%[^\n]s", str);

   l1 = strlen(str);

   printf("\nEnter a substring : ");
   scanf(" %[^\n]s", sub);

   l2 = strlen(sub);

   for (i = 0; i < l1;)
   {
      j = 0;
      count = 0;
      while ((str[i] == sub[j]))
      {
         count++;
         i++;
         j++;
      }
      if (count == l2)
```

```
    {
        count1++;
        count = 0;
    }
    else
        i++;
  }
      if(count1>=1)
  printf("%s is present in %s occurs %d", sub, str, count1);
  else
  printf("%s is absent in %s", sub, str);
}
```

Output:

```
Enter a string : trideep

Enter a substring : ee
ee is present in trideep occurs 1
--------------------------------
Process exited after 4.071 seconds with return value 33
Press any key to continue . . .
```

8.WAP to compare two strings without using library function.

Ans:

```
#include <stdio.h>
void compareStrings(char* x, char* y)
{
        int flag = 0;
        while (*x != '\0' || *y != '\0')
        {
                if (*x == *y) {
                        x++;
                        y++;
                }
                else if ((*x == '\0' && *y != '\0') || (*x != '\0' && *y == '\0') || *x != *y)
                {
                        flag = 1;
                        printf("Uequal Strings\n");
                        break;
                }
        }
        if (flag == 0)
        {
                printf("Equal Strings\n");
```

```
        }
}
int main(void)
{
        char s1[50],s2[50];
        printf("Enter two strings\n");
        scanf("%s",s1);
        scanf("%s",s1);
        compareStrings(s1, s2);
        return 0;
}
```

Output:

```
Enter two strings
trideep
trideeo
Uequal Strings

--------------------------------
Process exited after 4.589 seconds with return value 0
Press any key to continue . . .
```


Sagar Mohanty
Sic:180310003
CSE-C2
Roll-10