CSA0358 DATA STRUCTURES WITH GRAPH ALGORITHMS

DAY-2:(09/08/2023)

QUESTION 1:

Write a C program to calculate the sum of elements in an array.

CODE:

```
#include<stdio.h>
int main(){
	int n,i,sum=0;
	int a[n];
	printf("Enter the size of the array: ");
	scanf("%d",&n);
	printf("Enter the elements of the array: ");
	for(i=0;i<n;i++){
		scanf("%d",&a[i]);
	}
	for(i=0;i<n;i++){
		sum=sum+a[i];
	}
	printf("The sum of the elements in the array is %d",sum);
	return 0;
}
```

OUTPUT:

```
Enter the size of the array: 5
Enter the elements of the array: 2
3
4
5
6
The sum of the elements in the array is 20
--------------------------------
Process exited after 18.98 seconds with return value 0
Press any key to continue . . .
```

QUESTION 2:

Write a C program to merge two arrays.

CODE:

```
#include <stdio.h>
int main()
{
	int arr1size = 5, arr2size = 5, arr_resultsize, i, j;
	int a[5] = { 1, 2, 3, 4, 5 };

	int b[5] = { 6, 7, 8, 9, 10 };

	arr_resultsize = arr1size + arr2size;
	int c[arr_resultsize];

	for (i = 0; i < arr1size; i++) {
		c[i] = a[i];
	}

	for (i = 0, j = arr1size;
		j < arr_resultsize && i < arr2size; i++, j++) {
		c[j] = b[i];
	}

	for (i = 0; i < arr_resultsize; i++) {
		printf("%d ", c[i]);
	}
	return 0;
}
```

OUTPUT:

```
1 2 3 4 5 6 7 8 9 10
--------------------------------
Process exited after 3.974 seconds with return value 0
Press any key to continue . . .
```

QUESTION 3:

Write a C program to perform insertion, deletion of elements at the middle in an array.

CODE:

a).Insertion:

```
#include <stdio.h>
int main()
{
   int arr[100];
   int i, item, pos, size=7;
   printf("Enter 7 elements: ");
   for (i = 0; i < size; i++)
      scanf("%d",&arr[i]);
   printf("Array before insertion: ");
   for (i = 0; i < size; i++)
      printf("%d ", arr[i]);
   printf("\n");
   printf("Enter the element to be inserted: ");
   scanf("%d",&item);
   printf("Enter the position at which the element is to be inserted: ");
   scanf("%d",&pos);
   size++;
   for (i = size-1; i >= pos; i--)
      arr[i] = arr[i - 1];
   arr[pos - 1] = item;
   printf("Array after insertion: ");
      for (i = 0; i < size; i++)
      printf("%d ", arr[i]);
   printf("\n");
   return 0;
}
```

b).Deletion:

```
#include<stdio.h>
int main()
{
   int key, i, pos = -1, size=5;
   int arr[5] = {1, 20, 5, 78, 30};
   printf("Array before deletion: ");
   for (i = 0; i < size; i++)
      printf("%d ", arr[i]);
   printf("\n");
   printf("Enter element to delete: ");
```

```
    scanf("%d",&key);
    for(i = 0; i < size; i++)
    {
        if(arr[i] == key)
        {
            pos = i;
            break;
        }
    }
    if(pos != -1)
    {
        for(i = pos; i < size - 1; i++)
            arr[i] = arr[i+1];

        printf("Array after deletion: ");

        for(i = 0; i < size - 1; i++)
            printf("%d ",arr[i]);
    }
    else
        printf("Element Not Found\n");
    return 0;
}
```

OUTPUT:

a).Insertion:

```
Enter 7 elements: 1
2
3
4
5
6
7
Array before insertion: 1 2 3 4 5 6 7
Enter the element to be inserted: 8
Enter the position at which the element is to be inserted: 5
Array after insertion: 1 2 3 4 8 5 6 7

--------------------------------
Process exited after 15.36 seconds with return value 0
Press any key to continue . . .
```

b).Deletion:

```
Array before deletion: 1 20 5 78 30
Enter element to delete: 5
Array after deletion: 1 20 78 30
--------------------------------
Process exited after 4.507 seconds with return value 0
Press any key to continue . . .
```

QUESTION 4:

Write a C program to reverse a string.

CODE:

```
#include<stdio.h>
#include<string.h>
int main(){
        char s[]="libika";
        printf("The reversed string is %s",strrev(s));
        return 0;
}
```

OUTPUT:

```
The reversed string is akibil
--------------------------------
Process exited after 2.23 seconds with return value 0
Press any key to continue . . .
```

QUESTION 5:

Write a C program to check whether the string is palindrome or not palindrome.

CODE:

```
 #include <stdio.h>
#include <string.h>
int main()
{
   char string[25], reverse_string[25] = {'\0'};
   int  i, length = 0, flag = 0;
```

```
    fflush(stdin);
    printf("Enter a string: \n");
    gets(string);
    for (i = 0; string[i] != '\0'; i++)
    {
        length++;
    }
    for (i = length - 1; i >= 0; i--)
    {
       reverse_string[length - i - 1] = string[i];
    }
    for (i = 0; i < length; i++)
    {
        if (reverse_string[i] == string[i])
            flag = 1;
        else
            flag = 0;
            }
    if (flag == 1)
        printf("%s is a palindrome \n", string);
    else
        printf("%s is not a palindrome \n", string);
}
```

OUTPUT:

```
Enter a string:
libika
libika is not a palindrome

--------------------------------
Process exited after 4.132 seconds with return value 0
Press any key to continue . . .
```

QUESTION 6:

Write a C program to search a particular elements in a string.

CODE:

```
#include <stdio.h>
#include <string.h>
 int main()
```

```
{
   char s[1000],c;
   int i;
   printf("Enter  the string : ");
   gets(s);
   printf("Enter character to be searched: ");
   c=getchar();

   for(i=0;s[i];i++)
   {
        if(s[i]==c)
        {
                printf("character '%c' found at index: %d\n ",c,i);

              }
        }
   return 0;
}
```

OUTPUT:

```
Enter  the string : LIBIKA
Enter character to be searched: I
character 'I' found at index: 1
 character 'I' found at index: 3

--------------------------------
Process exited after 6.365 seconds with return value 0
Press any key to continue . . .
```

QUESTION 7:

**Write a C program to count number of times vowels(a,e,i,o,u)
Is present in the string.**

CODE:

```
#include <stdio.h>
int main()
{
  int c = 0, count = 0;
  char s[1000];
```

```
  printf("Input a string\n");
  gets(s);

  while (s[c] != '\0') {
    if (s[c] == 'a' || s[c] == 'A' || s[c] == 'e' || s[c] == 'E' || s[c] == 'i' || s[c] == 'I' ||
s[c] =='o' || s[c]=='O' || s[c] == 'u' || s[c] == 'U')
      count++;
    c++;
  }

  printf("Number of vowels in the string: %d", count);

  return 0;
}
```

OUTPUT:

```
Enter  the string : LIBIKA
Enter character to be searched: I
character 'I' found at index: 1
 character 'I' found at index: 3

--------------------------------
Process exited after 6.365 seconds with return value 0
Press any key to continue . . .
```

QUESTION 8:

Write a C program to calculate Matrix Multiplication.

CODE:

```
#include<stdio.h>
#include<stdlib.h>
int main(){
int a[10][10],b[10][10],mul[10][10],r,c,i,j,k;
system("cls");
printf("enter the number of row=");
scanf("%d",&r);
printf("enter the number of column=");
scanf("%d",&c);
printf("enter the first matrix element=\n");
for(i=0;i<r;i++)
```

```
{
for(j=0;j<c;j++)
{
scanf("%d",&a[i][j]);
}
}
printf("enter the second matrix element=\n");
for(i=0;i<r;i++)
{
for(j=0;j<c;j++)
{
scanf("%d",&b[i][j]);
}
}

printf("multiply of the matrix=\n");
for(i=0;i<r;i++)
{
for(j=0;j<c;j++)
{
mul[i][j]=0;
for(k=0;k<c;k++)
{
mul[i][j]+=a[i][k]*b[k][j];
}
}
}
//for printing result
for(i=0;i<r;i++)
{
for(j=0;j<c;j++)
{
printf("%d\t",mul[i][j]);
}
printf("\n");
}
return 0;
}
```

OUTPUT:

```
enter the number of row=2
enter the number of column=2
enter the first matrix element=
1
2
3
4
enter the second matrix element=
5
6
7
8
multiply of the matrix=
19      22
43      50

--------------------------------
Process exited after 15.19 seconds with return value 0
Press any key to continue . . .
```

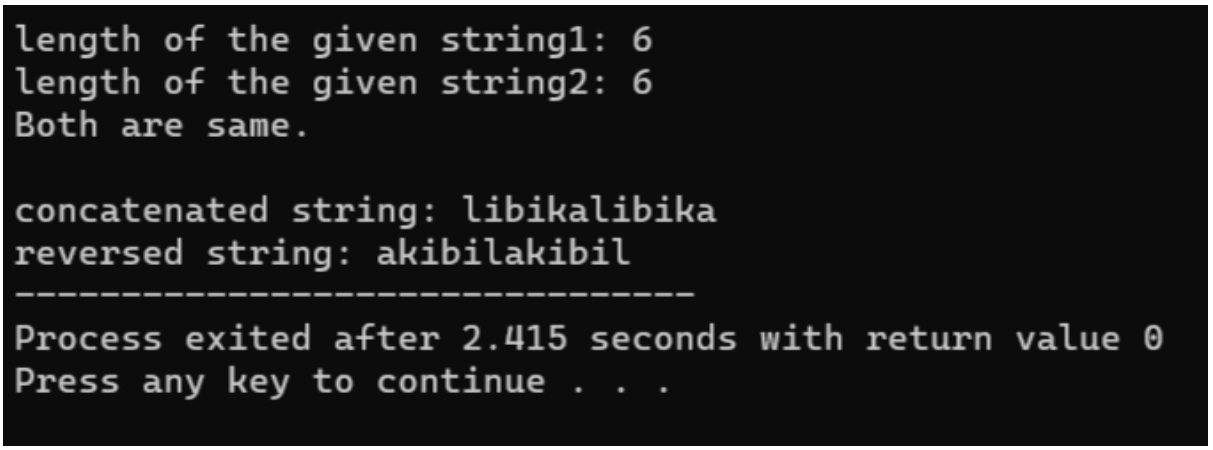
QUESTION 9:

Write a C program to perform all string manipulation operations in a string.

CODE:

```
#include<stdio.h>
#include<string.h>
int main()
{
        char s[]="libika",s1[]="libika";
        printf("length of the given string1: %d",strlen(s));
        printf("\nlength of the given string2: %d",strlen(s1));
        if (strcmp(s, s1) == 0) {
   printf("\nBoth are same.\n");
        }
        else {
        printf("Both are different.\n");
        }
        printf("\nconcatenated string: %s",strcat(s,s1));
        printf("\nreversed string: %s",strrev(s));
        return 0;
}
```

OUTPUT:


length of the given string1: 6
length of the given string2: 6
Both are same.
concatenated string: libikalibika
reversed string: akibilakibil
Process exited after 2.415 seconds with return value 0
Press any key to continue . . .