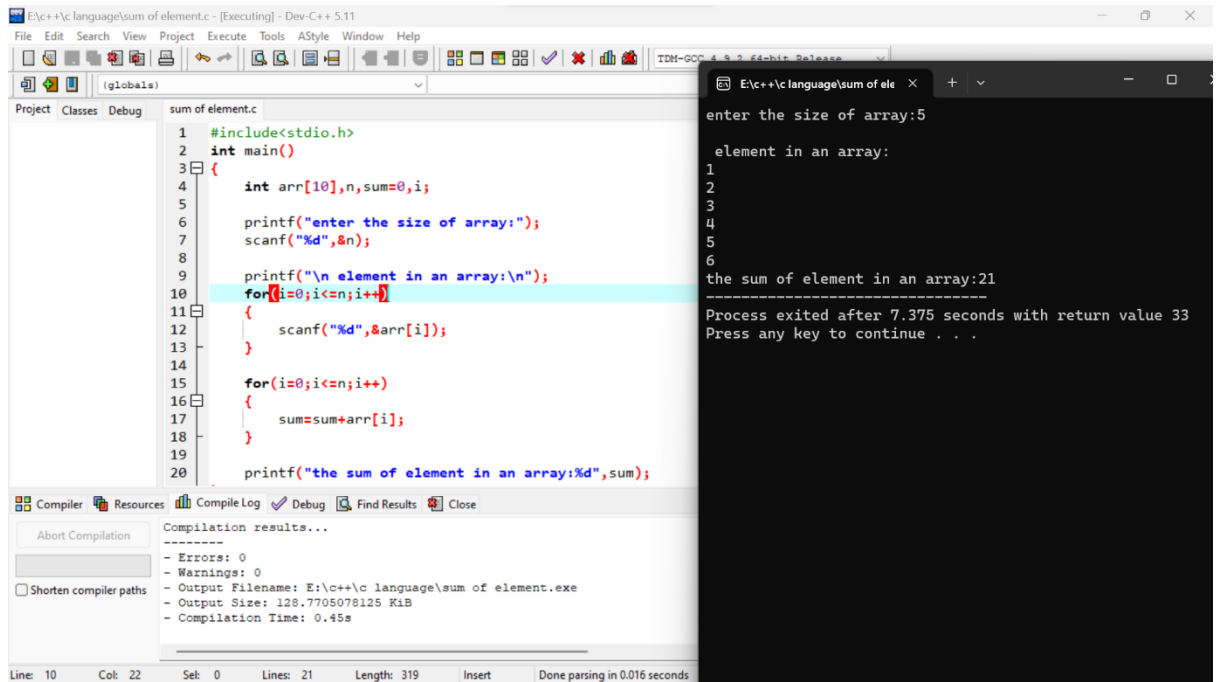


DAY 2

DATA STRUCTURES – C PROGRAMMING – 2

1. Write a c program to calculate sum of elements in an array.

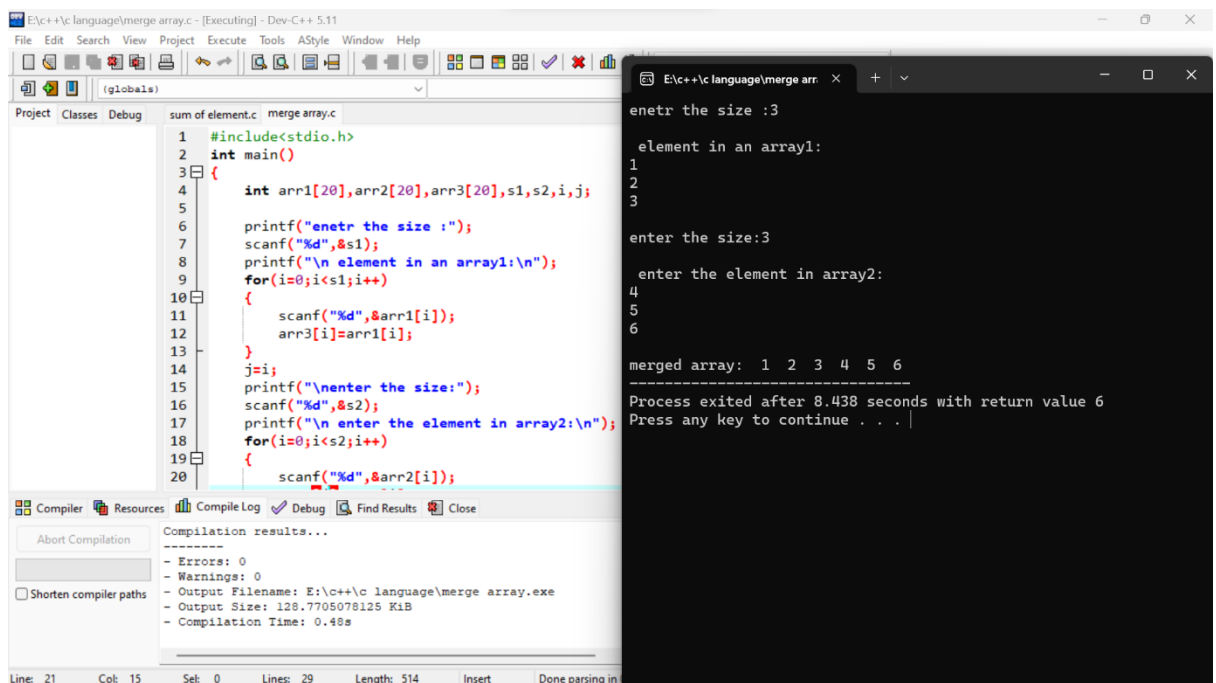


```
#include<stdio.h>
int main()
{
    int arr[10],n,sum=0,i;
    printf("enter the size of array:");
    scanf("%d",&n);
    printf("\n element in an array:\n");
    for(i=0;i<=n;i++)
    {
        scanf("%d",&arr[i]);
    }
    for(i=0;i<=n;i++)
    {
        sum=sum+arr[i];
    }
    printf("the sum of element in an array:%d",sum);
}
```

enter the size of array:5
element in an array:
1
2
3
4
5
6
the sum of element in an array:21

Process exited after 7.375 seconds with return value 33
Press any key to continue . . .

2. Write a c program to merge two array.



```
#include<stdio.h>
int main()
{
    int arr1[20],arr2[20],arr3[20],s1,s2,i,j;
    printf("enetr the size :");
    scanf("%d",&s1);
    printf("\n element in an array1:\n");
    for(i=0;i<s1;i++)
    {
        scanf("%d",&arr1[i]);
        arr3[i]=arr1[i];
    }
    j=i;
    printf("\nenter the size:");
    scanf("%d",&s2);
    printf("\n enter the element in array2:\n");
    for(i=0;i<s2;i++)
    {
        scanf("%d",&arr2[i]);
    }
}
```

enetr the size :3
element in an array1:
1
2
3
enter the size:3
enter the element in array2:
4
5
6
merged array: 1 2 3 4 5 6

Process exited after 8.438 seconds with return value 6
Press any key to continue . . .

3. Write a c program to perform insertion and deletion of the elements in the middle of the array.

The screenshot shows the Dev-C++ IDE with a project named "insert and delete in middle.c". The code in the editor is as follows:

```
1 #include<stdio.h>
2 int main()
3 {
4     int arr[10];
5     int n,pos,value,i;
6
7     printf("enter the no of element:");
8     scanf("%d",&n);
9
10    printf("\n enter the element:");
11    for(i=0;i<n;i++)
12    {
13        scanf("%d",&arr[i]);
14    }
15
16    printf("\n enter the position:");
17    scanf("%d",&pos);
18
19    switch(pos)
20    {
21
22    case 1:
23        if(pos>=3 && pos<=n+1)
24        {
25            for(i=n-1;i>=pos-1;i--)
26                ...
```

The output console shows the following interaction:

```
enter the no of element:6
enter the element:1
2
3
4
5
6
enter the position:4
enter the value::5
element inserted
updated array: 1 2 3 5 4 5 6
-----
Process exited after 10.26 seconds with return value 7
Press any key to continue . . . |
```

4. Write a c program to reverse a string.

The screenshot shows the Dev-C++ IDE with a project named "rev string.c". The code in the editor is as follows:

```
1 #include<stdio.h>
2 #include<string.h>
3 int main()
4 {
5     char str[40];
6
7     printf("string before reverse:");
8     scanf("%s",&str);
9
10    printf("string after reversed:%s",strrev(str));
11
12 }
```

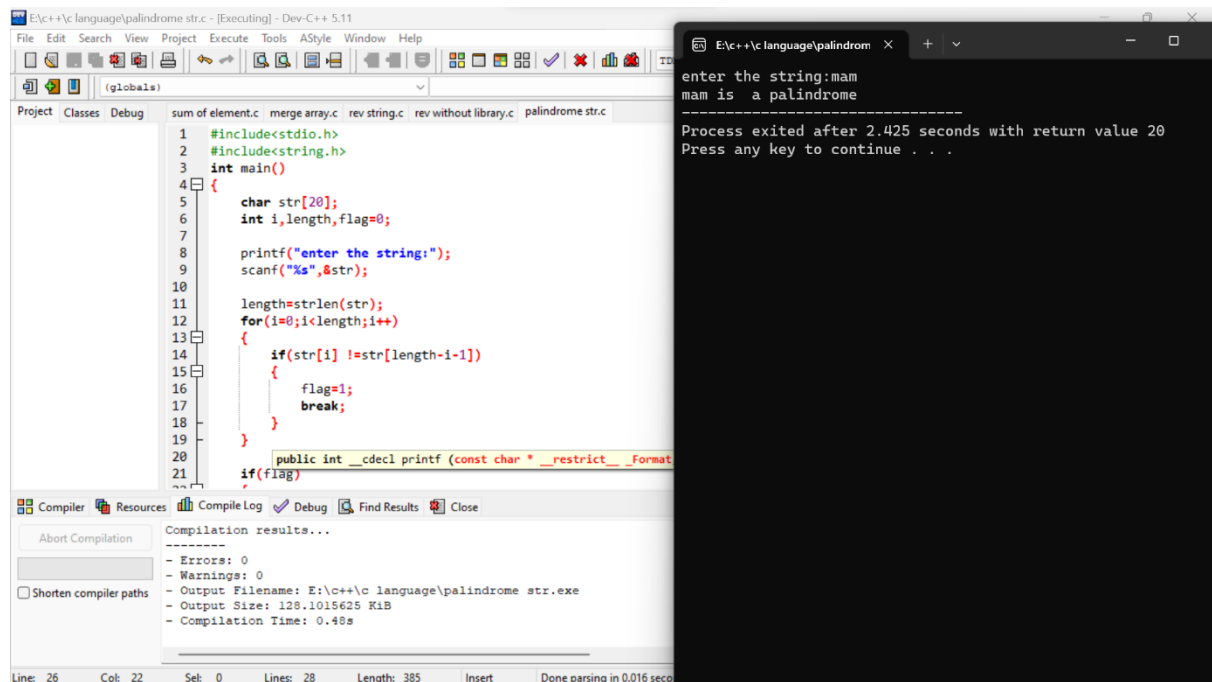
The output console shows the following interaction:

```
string before reverse:sriram
string after reversed:marisr
-----
Process exited after 9.497 seconds with return value 28
Press any key to continue . . . |
```

The compilation results at the bottom of the IDE are as follows:

```
Compilation results...
-----
- Errors: 0
- Warnings: 0
- Output Filename: E:\c++\c language\rev string.exe
- Output Size: 128.7724609375 KiB
- Compilation Time: 0.47s
```

5. Write a c program to check the given string is palindrome or not.



```
1 #include<stdio.h>
2 #include<string.h>
3 int main()
4 {
5     char str[20];
6     int i,length,flag=0;
7
8     printf("enter the string:");
9     scanf("%s",&str);
10
11     length=strlen(str);
12     for(i=0;i<length;i++)
13     {
14         if(str[i] !=str[length-i-1])
15         {
16             flag=1;
17             break;
18         }
19     }
20     printf("The string is %s", flag==0?"palindrome":"not a palindrome");
21 }
```

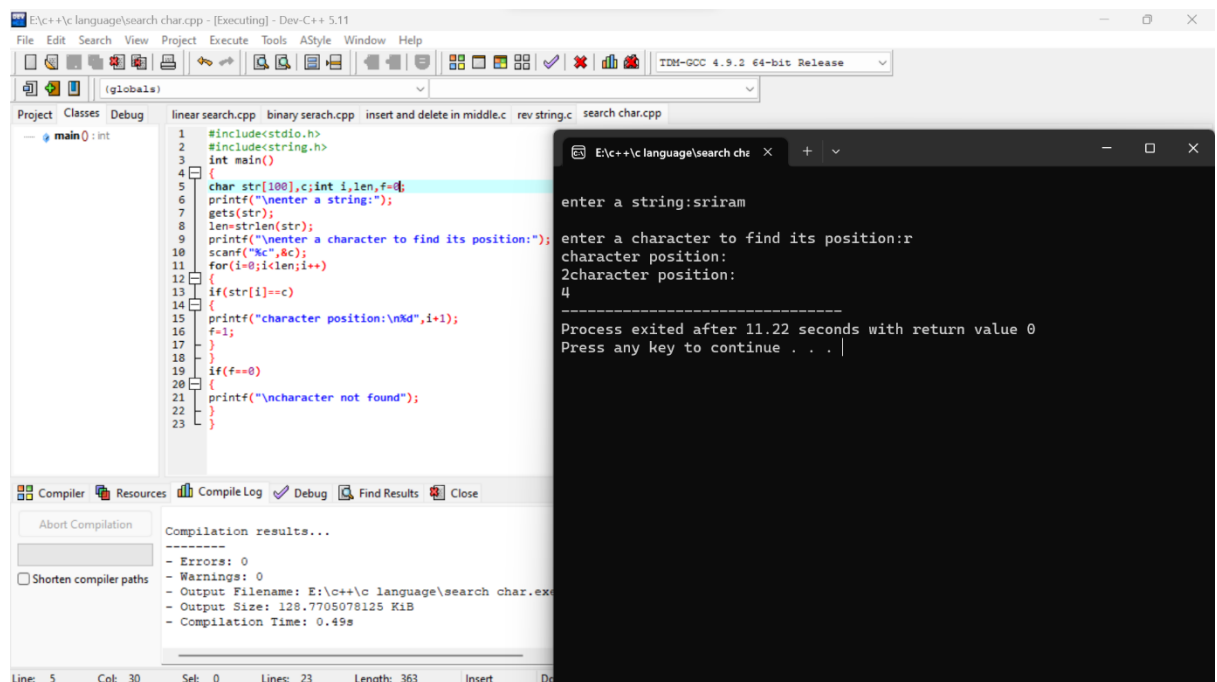
Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: E:\c++\c language\palindrome str.exe
- Output Size: 128.1015625 KiB
- Compilation Time: 0.48s

enter the string:mam
mam is a palindrome

Process exited after 2.425 seconds with return value 20
Press any key to continue . . .

6. Write a c program to search for a particular character in a string.



```
1 #include<stdio.h>
2 #include<string.h>
3 int main()
4 {
5     char str[100];int i,len,f=0;
6     printf("\nEnter a string:");
7     gets(str);
8     len=strlen(str);
9     printf("\nEnter a character to find its position:");
10    scanf("%c",&c);
11    for(i=0;i<len;i++)
12    {
13        if(str[i]==c)
14        {
15            printf("character position:\n%d",i+1);
16            f=1;
17        }
18    }
19    if(f==0)
20    {
21        printf("\ncharacter not found");
22    }
23 }
```

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: E:\c++\c language\search char.exe
- Output Size: 128.7705078125 KiB
- Compilation Time: 0.49s

enter a string:sriram
enter a character to find its position:r
character position:
2character position:
4

Process exited after 11.22 seconds with return value 0
Press any key to continue . . .

7. Write a c program to count the number of lines a,e,l,o,u present in the string.

```

1 #include<stdio.h>
2 int main()
3 {
4     int c=0,count=0;
5     char s[100];
6
7     printf("enter the string:");
8     scanf("%s",&s);
9
10    while(s[c] != '\0')
11    {
12        if((s[c]=='a' || s[c]=='e' || s[c]=='i' || s[c]=='o' || s[c]=='u' ||
13            s[c]=='A' || s[c]=='E' || s[c]=='I' || s[c]=='O' || s[c]=='U'))
14            count++;
15        c++;
16    }
17    printf("number of vowels:%d",count);
18 }

```

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: E:\c++\c language\count vowel.exe
- Output Size: 128.1015625 KiB
- Compilation Time: 0.49s

enter the string:sriram
number of vowels:2

Process exited after 7.097 seconds with return value 0
Press any key to continue . . .

8. write a c program to perform matrix multiplication.

```

1 #include<stdio.h>
2 #include<stdlib.h>
3 int main()
4 {
5     int a[10][10],b[10][10],mul[10][10],r,c,i,j,k;
6     system("cls");
7     printf("enter the number of row=");
8     scanf("%d",&r);
9     printf("enter the number of column=");
10    scanf("%d",&c);
11    printf("enter the first matrix element\n");
12    for(i=0;i<r;i++)
13    {
14        for(j=0;j<c;j++)
15        {
16            scanf("%d",&a[i][j]);
17        }
18    }
19    printf("enter the second matrix element\n");
20    for(i=0;i<r;i++)
21    {
22        for(j=0;j<c;j++)
23        {
24            scanf("%d",&b[i][j]);
25        }
26    }

```

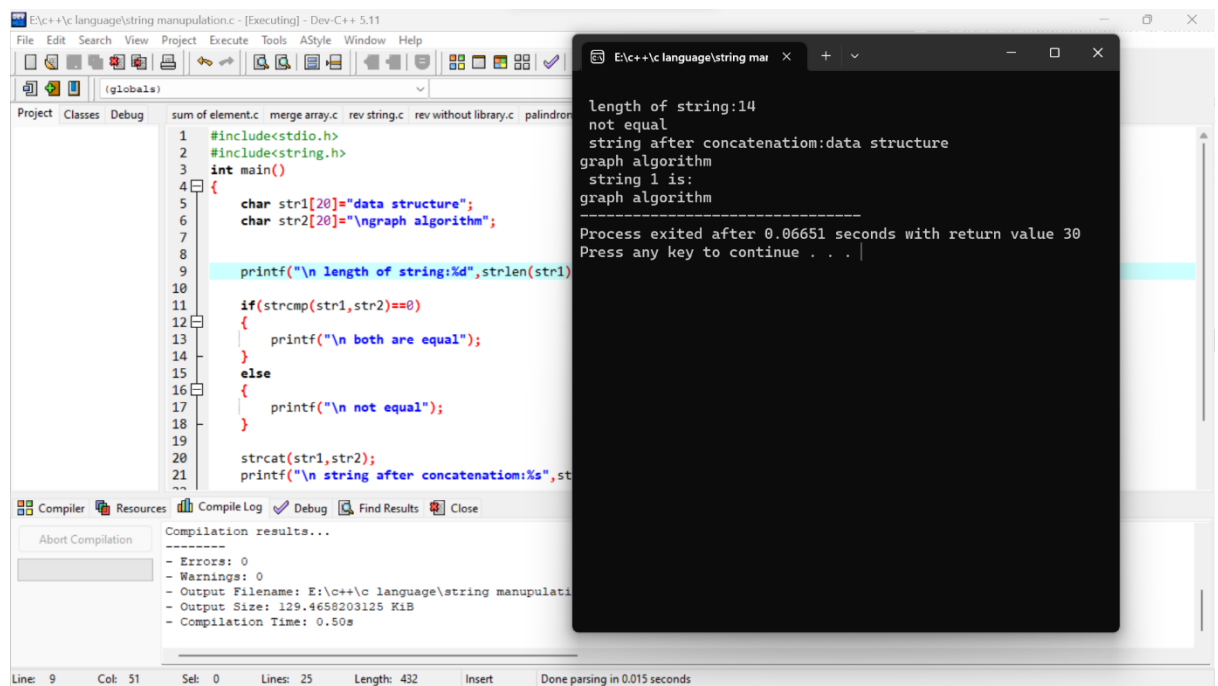
Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: E:\c++\c language\matrix mul.exe
- Output Size: 130.11320125 KiB
- Compilation Time: 0.59s

enter the number of row=3
enter the number of column=3
enter the first matrix element=
1 2 3
4 5 6
7 8 9
enter the second matrix element=
3 4 7
8 6 2
9 5 1
multiply of the matrix=
46 31 14
106 76 44
166 121 74

Process exited after 33.08 seconds with return value 0
Press any key to continue . . .

9. write a c program to perform all string manipulation.



The screenshot displays the Dev-C++ IDE with a C program for string manipulation. The program defines two strings, "data structure" and "graph algorithm", compares them, and concatenates them. A console window shows the program's output, including the length of the first string, a comparison result, and the concatenated string.

```
#include<stdio.h>
#include<string.h>
int main()
{
    char str1[20]="data structure";
    char str2[20]="\ngraph algorithm";

    printf("\n length of string:%d",strlen(str1));

    if(strcmp(str1,str2)==0)
    {
        printf("\n both are equal");
    }
    else
    {
        printf("\n not equal");
    }

    strcat(str1,str2);
    printf("\n string after concatenation:%s",str1);
}
```

length of string:14
not equal
string after concatenation:data structure
graph algorithm
string 1 is:
graph algorithm

Process exited after 0.06651 seconds with return value 30
Press any key to continue . . .

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: E:\c++\c language\string manipulation.exe
- Output Size: 129.4658203125 KiB
- Compilation Time: 0.50s