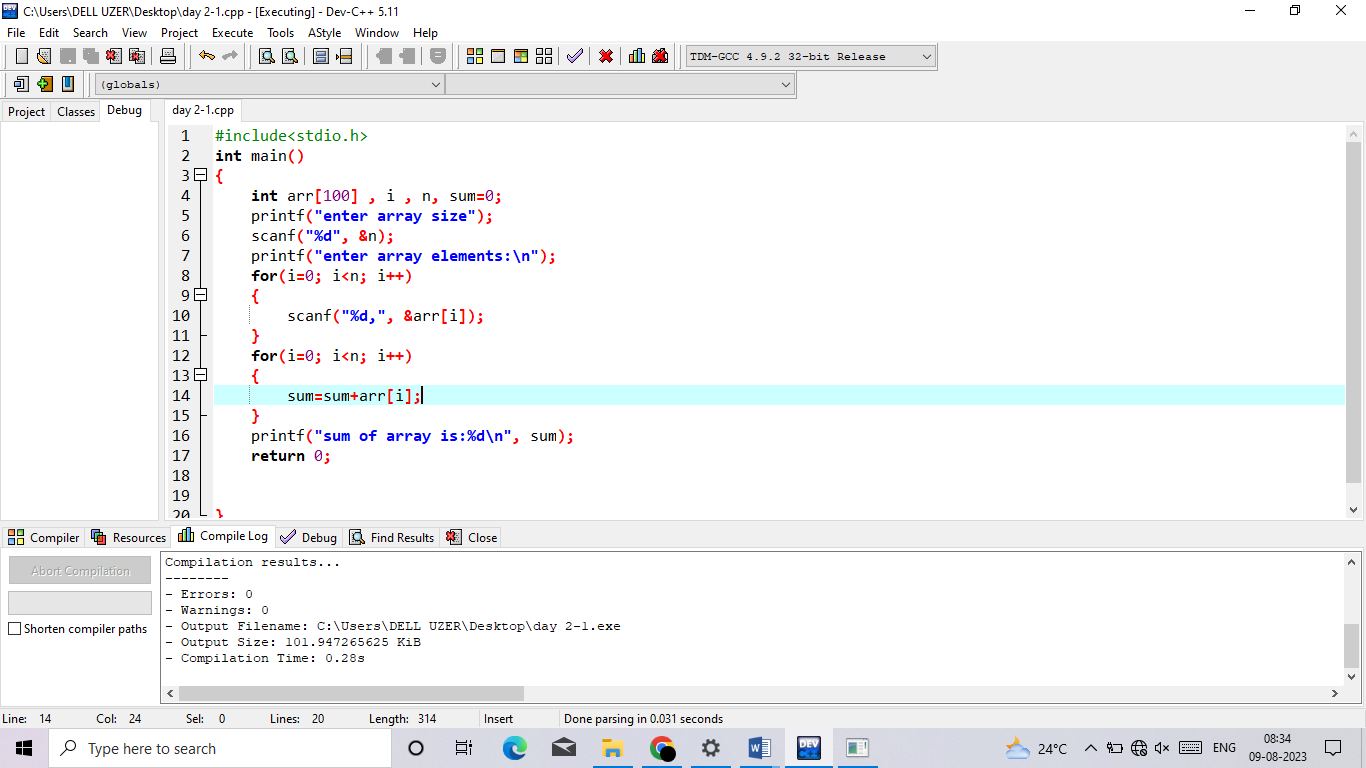
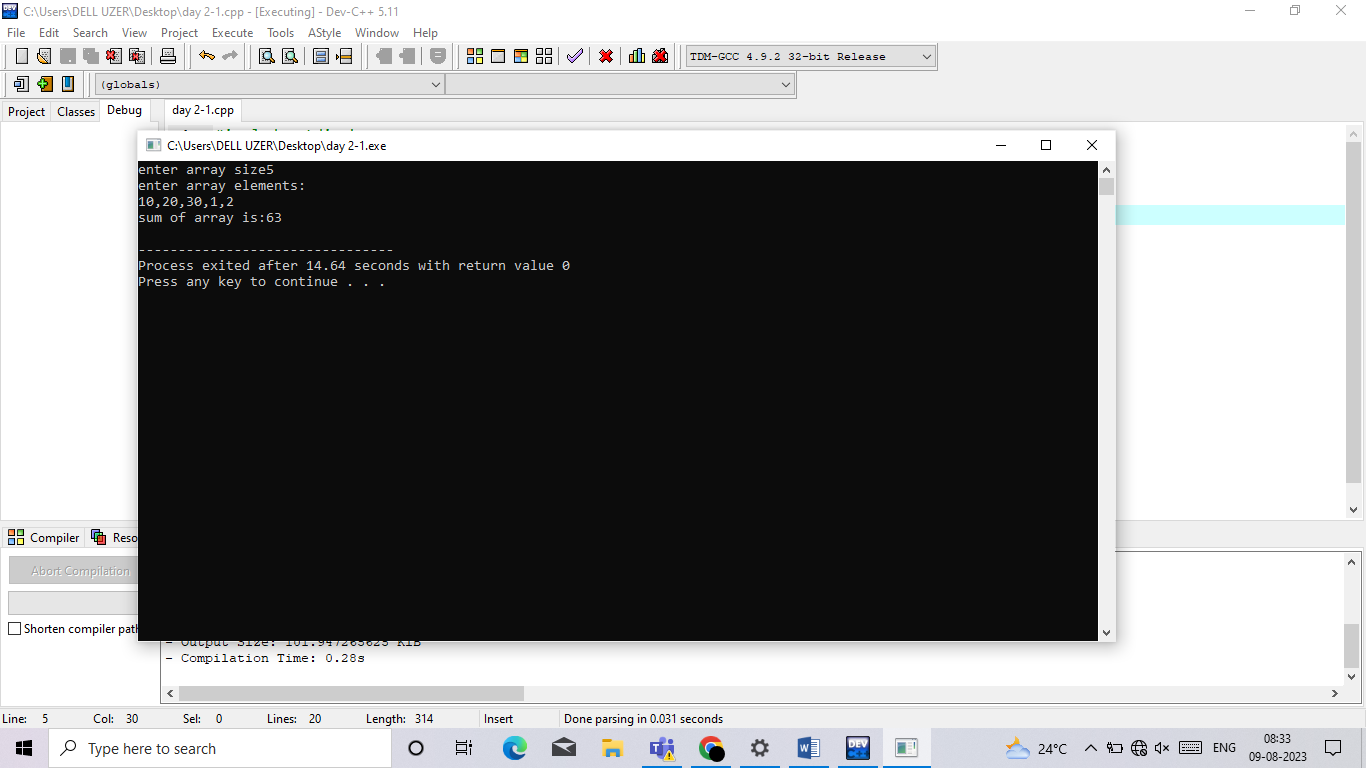
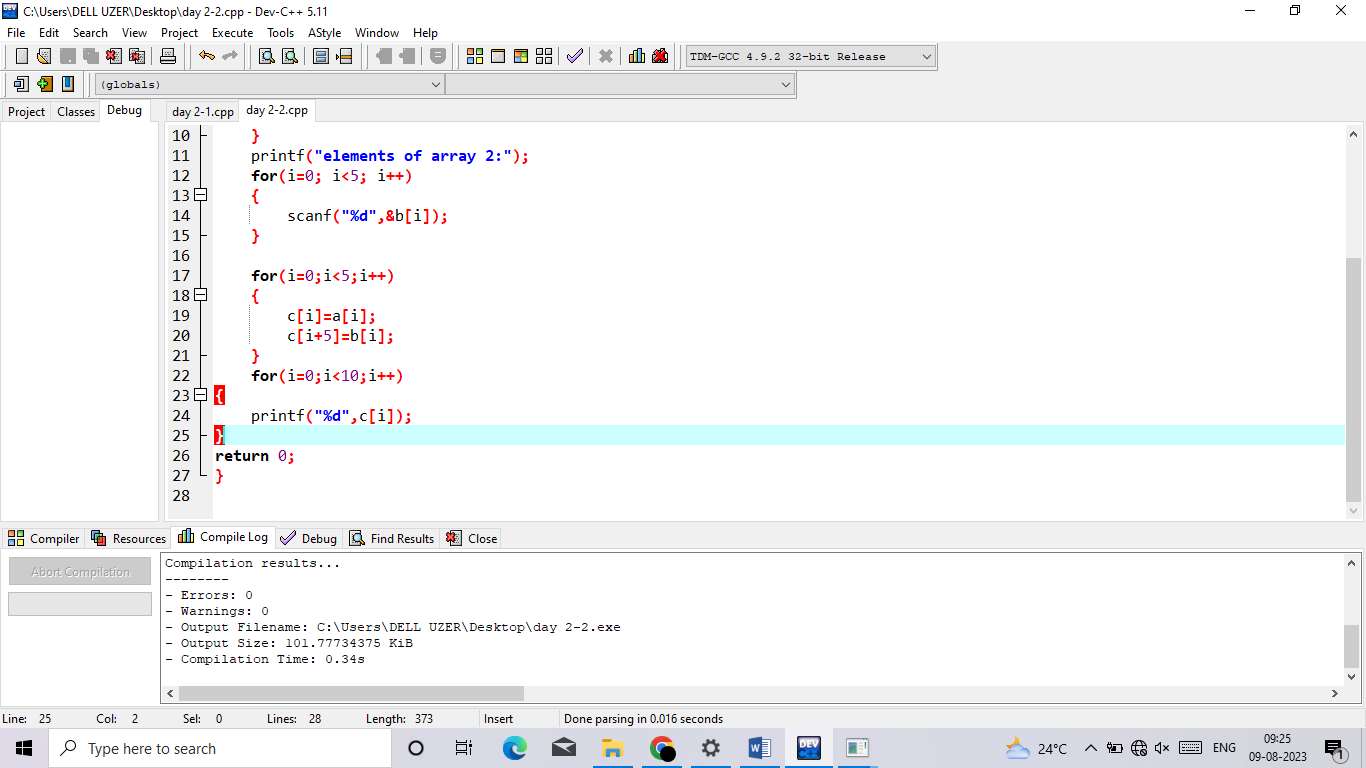
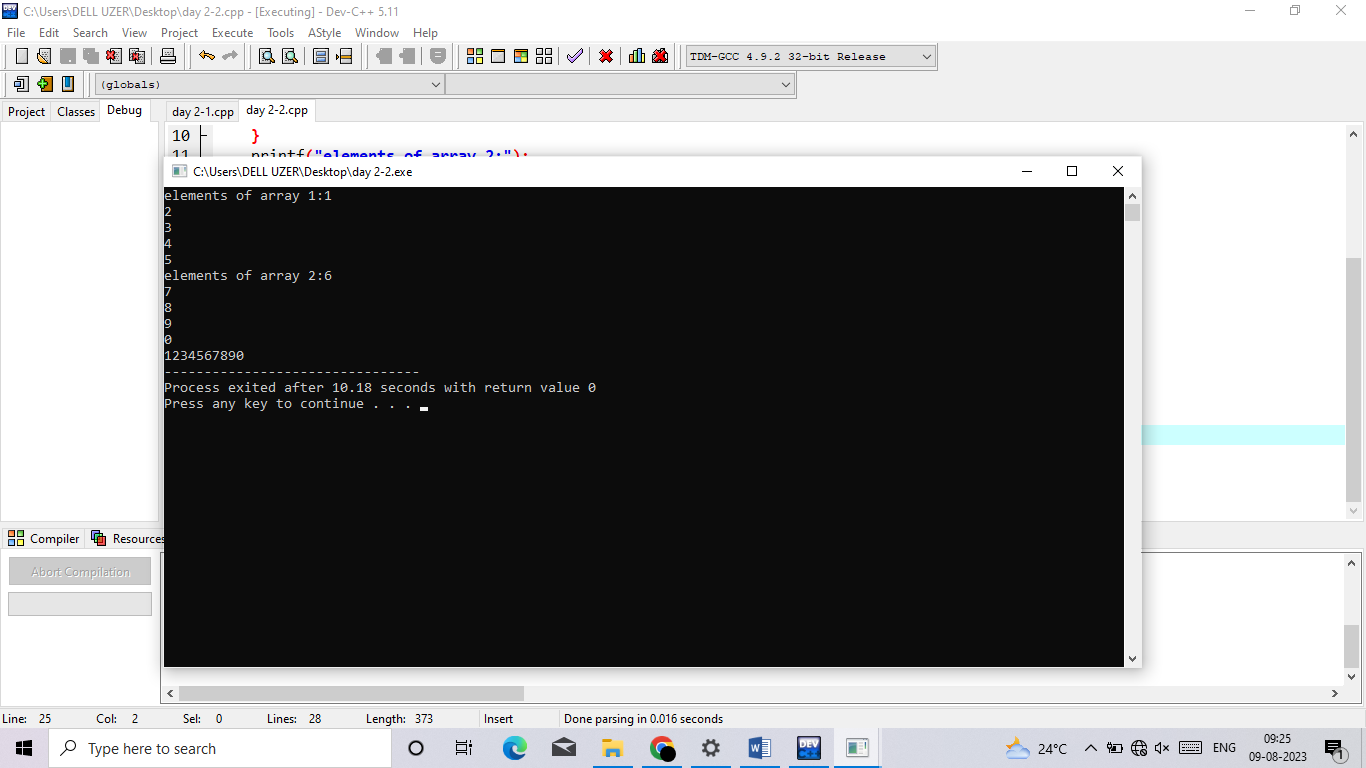
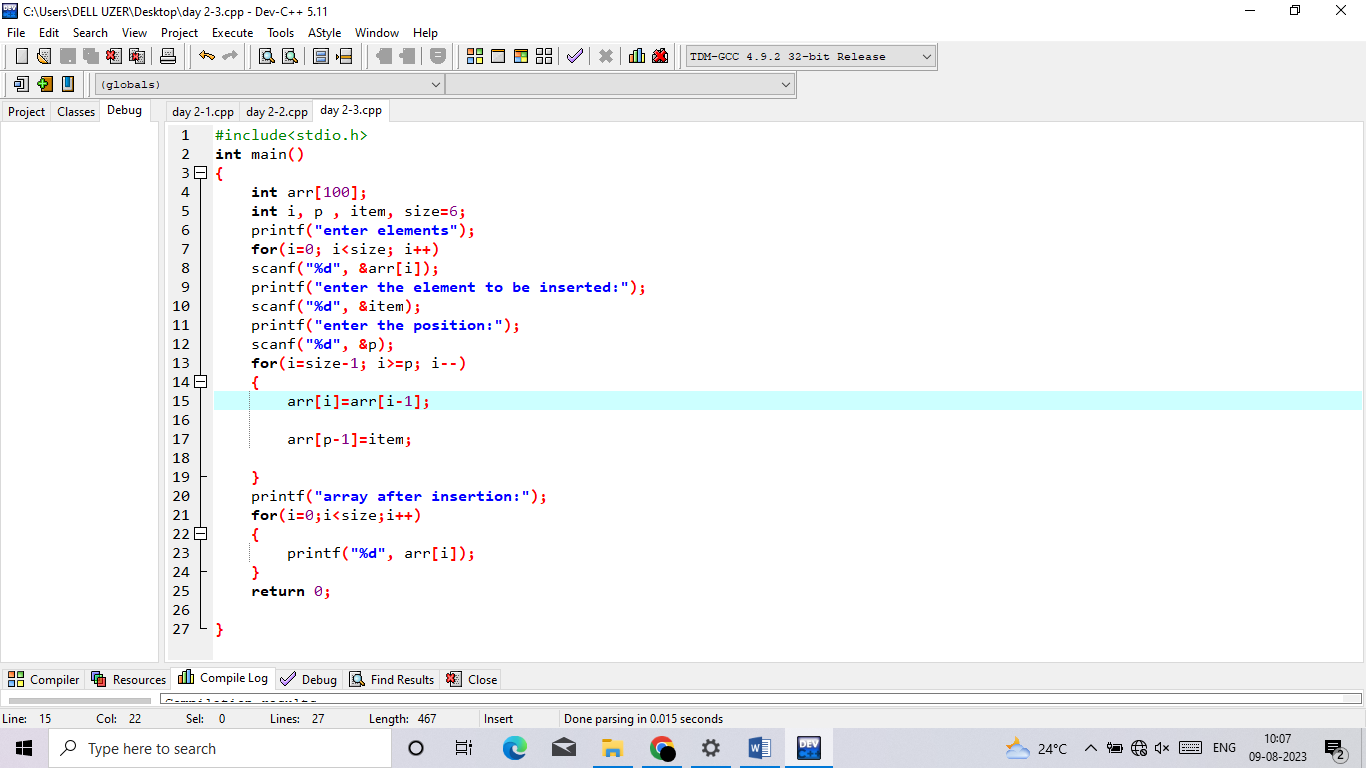
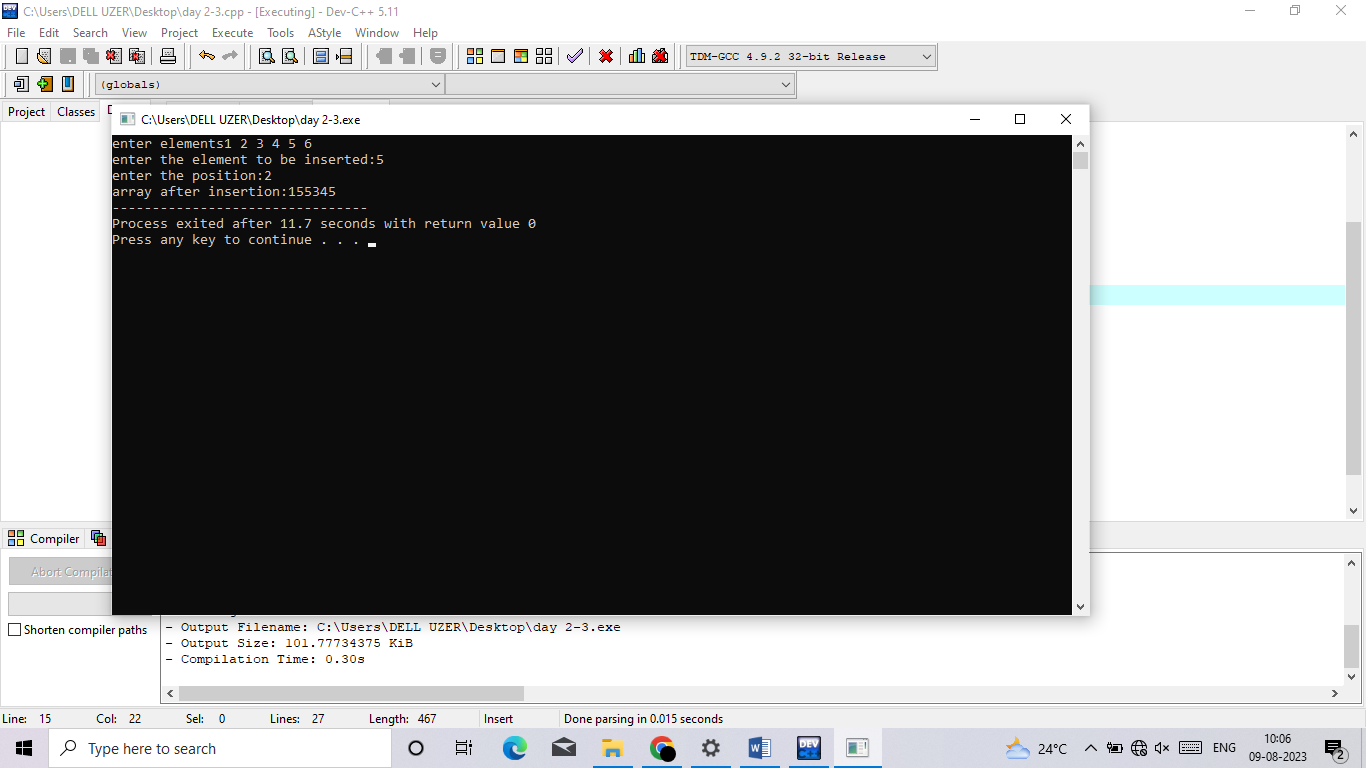
**Data structures day 2**

1. Sum of elements in a array
2. Merge 2 arrays

3. to perform insertion and deletion in middle array



Deletion:

#include <stdio.h>

#define MAX\_SIZE 100

int main() {

int array[MAX\_SIZE];

int n, pos;

printf("Enter the number of elements in the array: ");

scanf("%d", &n);

printf("Enter the elements:\n");

for (int i = 0; i < n; i++) {

scanf("%d", &array[i]);

}

printf("Enter the position of the element to be deleted: ");

scanf("%d", &pos);

switch (pos) {

case 1:

for (int i = 0; i < n - 1; i++) {

array[i] = array[i + 1];

}

n--;

printf("Element deleted successfully.\n");

break;

case 2:

n--;

printf("Element deleted successfully.\n");

break;

default:

if (pos >= 3 && pos <= n) {

for (int i = pos - 1; i < n - 1; i++) {

array[i] = array[i + 1];

}

n--;

printf("Element deleted successfully.\n");

} else {

printf("Invalid position.\n");

}

break;

}

printf("Updated array: ");

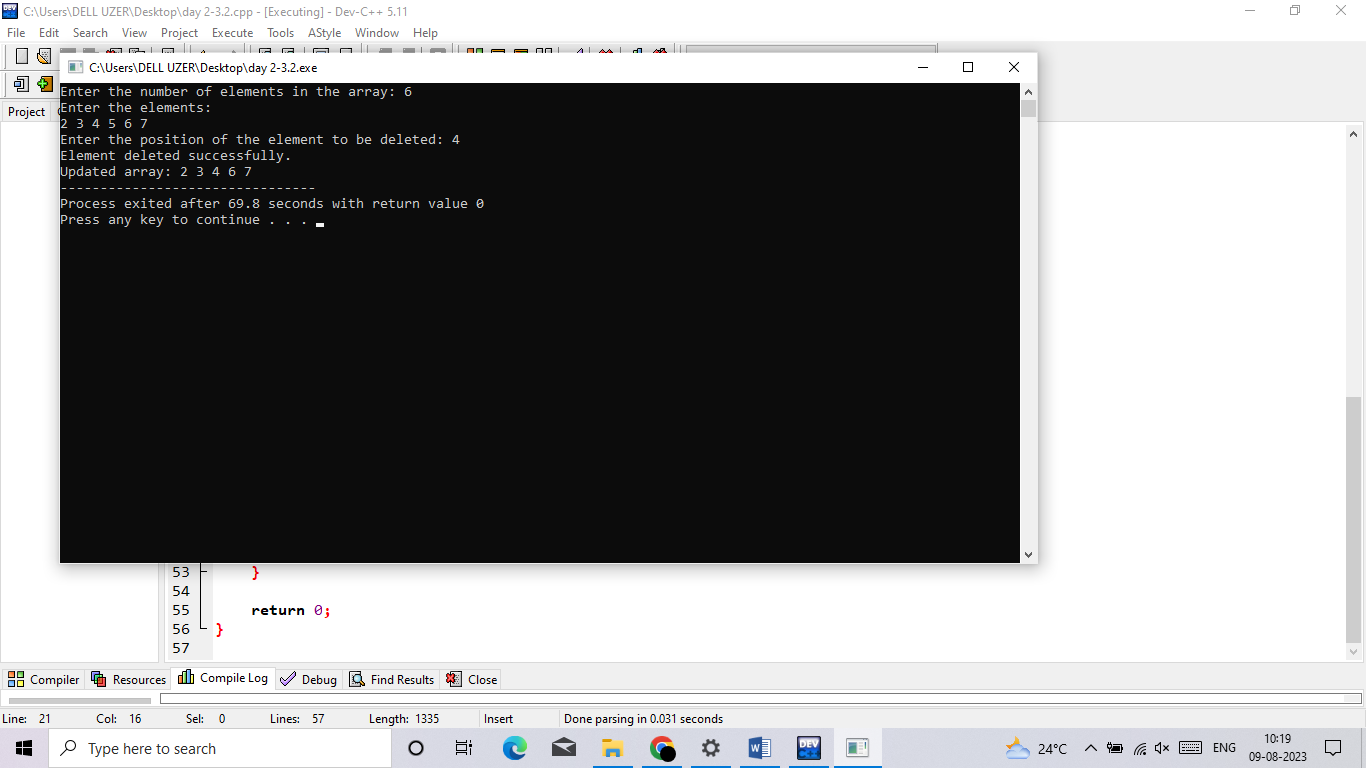
for (int i = 0; i < n; i++) {

printf("%d ", array[i]);

}

return 0;

}



4 reverse a string

#include <stdio.h>

#include <string.h>

int main() {

char str[100];

printf("Enter a string:");

scanf("%s", str); // read string

int length = strlen(str);

int start=0, end=length-1;

while (start < end) {

char temp = str[start];

str[start] = str[end];

str[end] = temp;

start++;

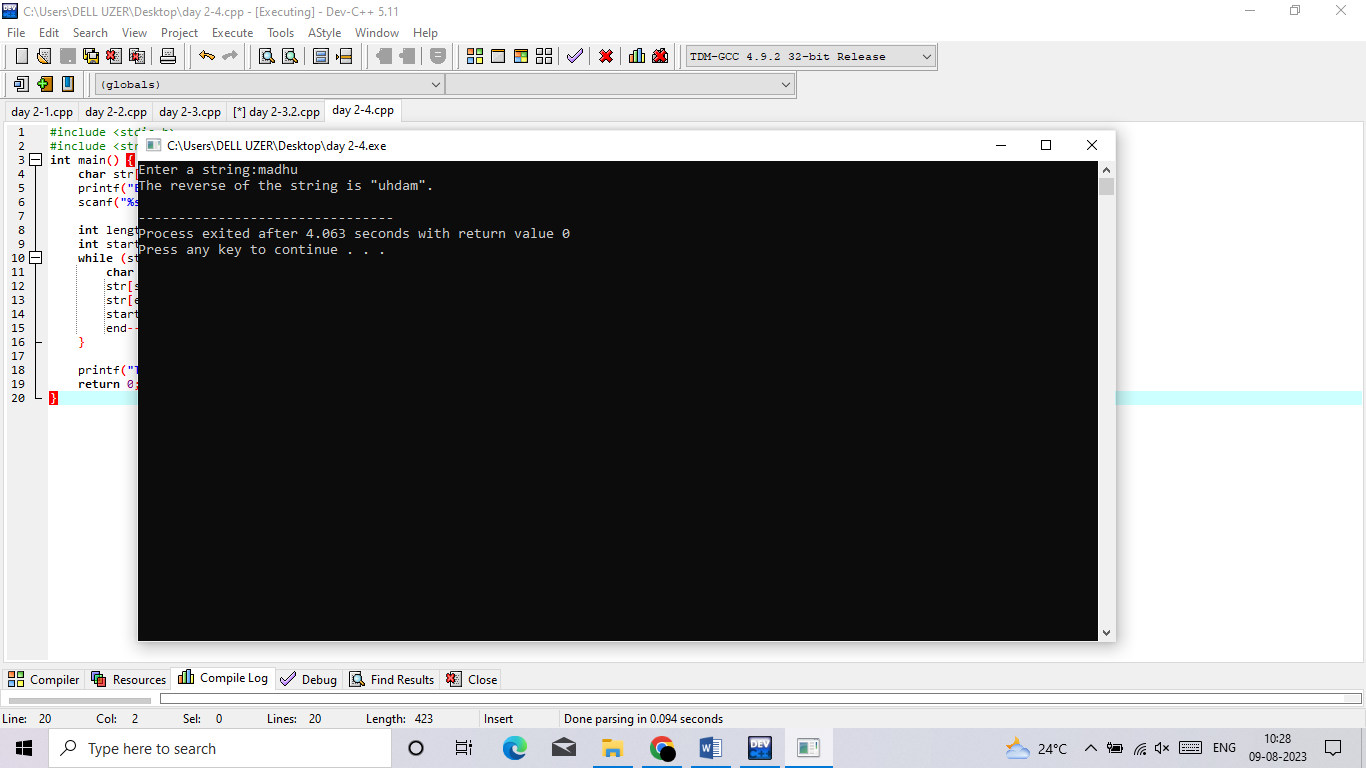
end--;

}

printf("The reverse of the string is \"%s\".\n", str);

return 0;

}



5 palindrome

#include <stdio.h>

#include <string.h>

int main()

{

char str[10] = "madhu";

int i, len, flag = 0;

len = strlen(str);

for (i = 0; i < len; i++)

{

// Checking if string is palindrome or not

if (str[i] != str[len - i - 1]) {

flag = 1;

break;

}

}

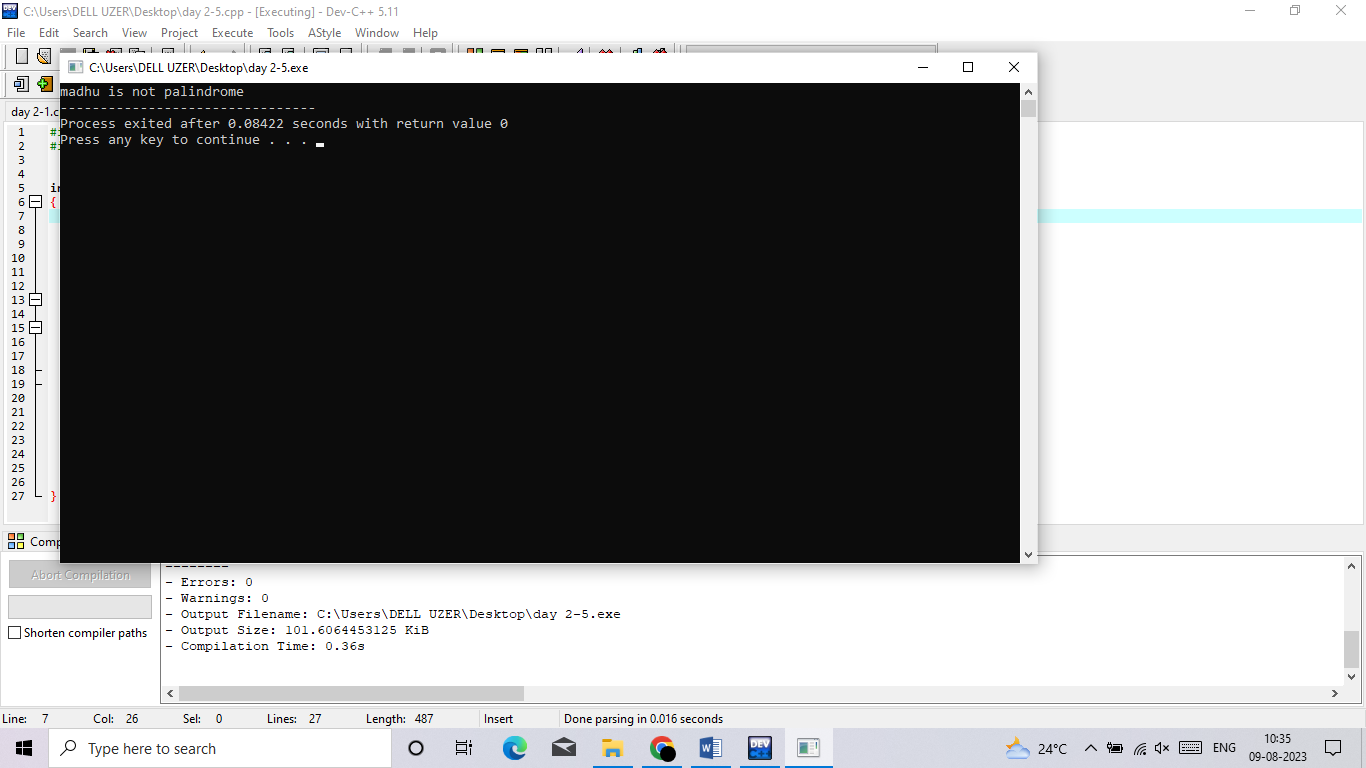
if (flag)

printf("%s is not palindrome", str);

else

printf("%s is palindrome", str);

return 0;

}

6 to search a particular character in a string

#include<stdio.h>

#include<string.h>

int main()

{

char str[100],c;int i,len, f=0;

printf("enter a string:");

gets(str);

len=strlen(str);

printf("enter a character to find its position:");

scanf("%c",&c);

for(i=0;i<len;i++)

{

if(str[i]==c)

{

printf("character position:%d",i+1);

f=1;

}

}

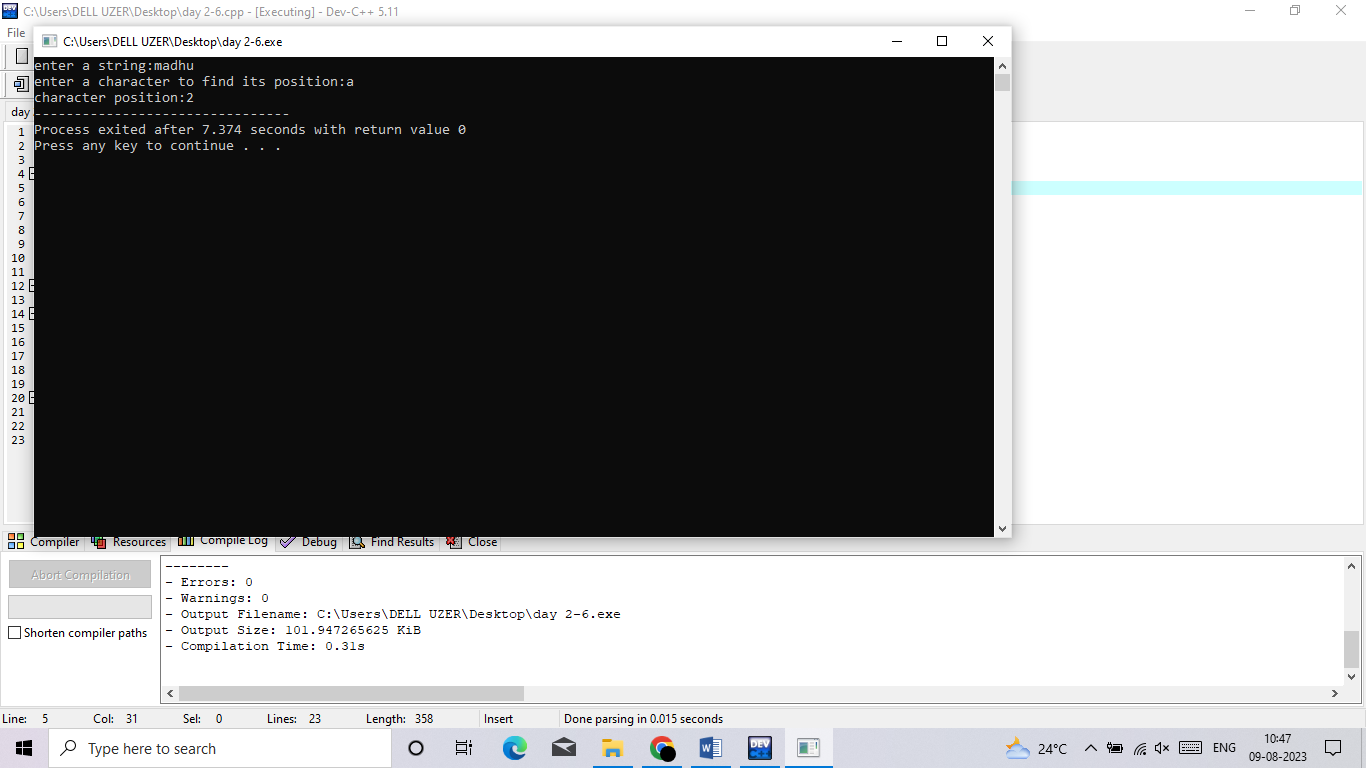
if(f==0)

{

printf("\ncharacter not found");

}

}



7 count no. of vowels present

#include <stdio.h>

#include <ctype.h>

#include <string.h>

int isVowel(char ch)

{

ch = toupper(ch);

return (ch == 'A' || ch == 'E' || ch == 'I' || ch == 'O' || ch == 'U');

}

int countVowel(char str[], int n)

{

if (n == 1)

return isVowel(str[n - 1]);

return countVowel(str, n - 1) + isVowel(str[n - 1]);

}

int main() {

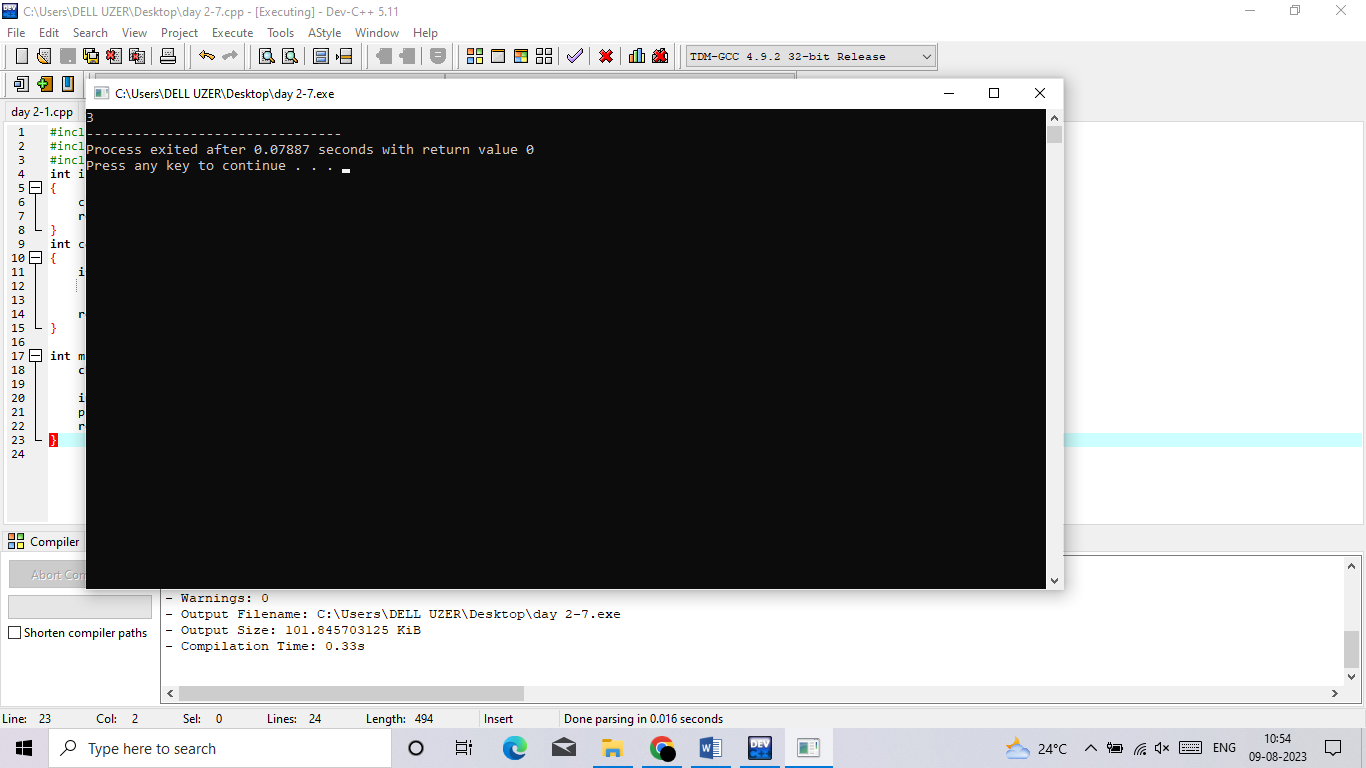
char str[10] = "Hey Dude";

int len = strlen(str);

printf("%d", countVowel(str, len));

return 0;

}



8. matrix multiplication

#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]);

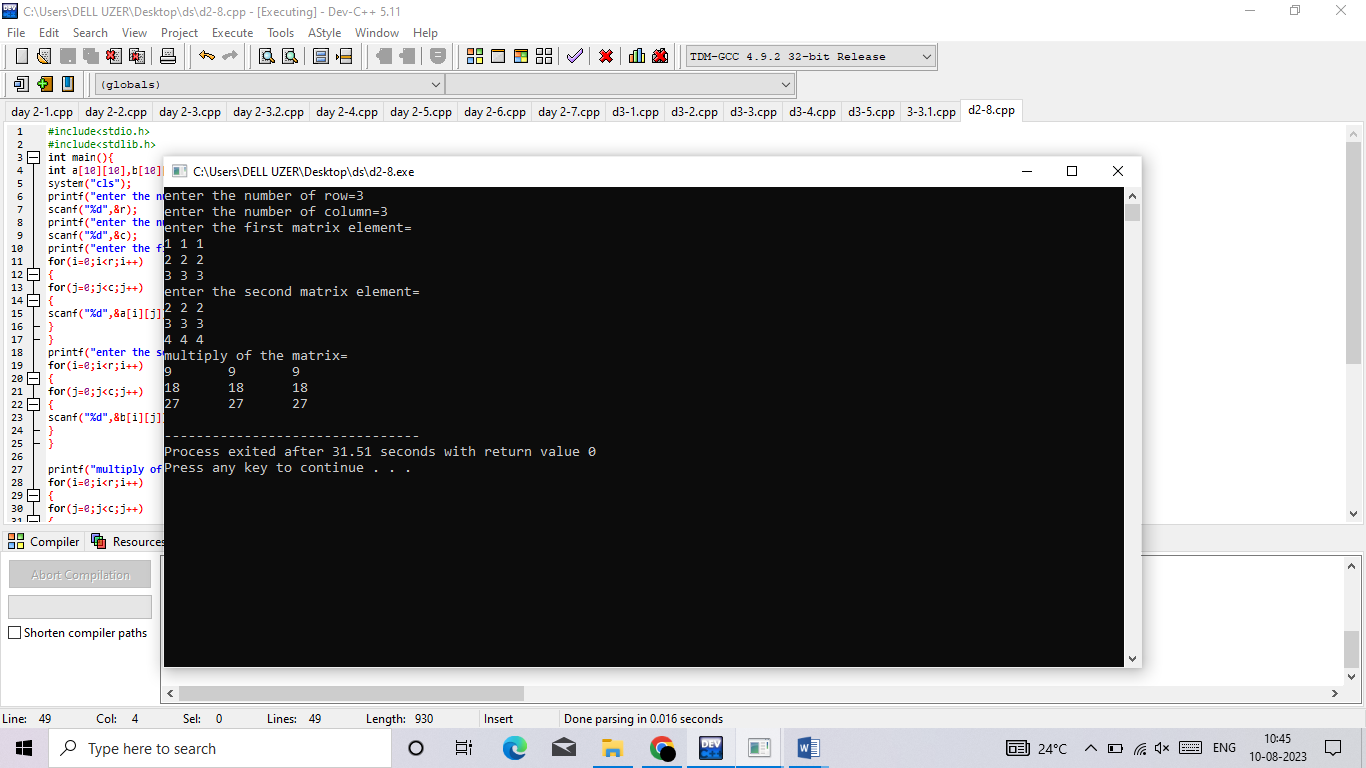
}

printf("\n");

}

return 0;

}



9. concatenated string

#include <stdio.h>

int main()

{

char first\_string[20];

char second\_string[20];

int i;

printf("Enter the first string");

scanf("%s",first\_string);

printf("\nEnter the second string");

scanf("%s",second\_string);

for(i=0;first\_string[i]!='\0';i++);

for(int j=0;second\_string[j]!='\0';j++)

{

first\_string[i]=second\_string[j];

i++;

}

first\_string[i]='\0';

printf("After concatenation, the string would look like: %s", first\_string);

return 0;

}

