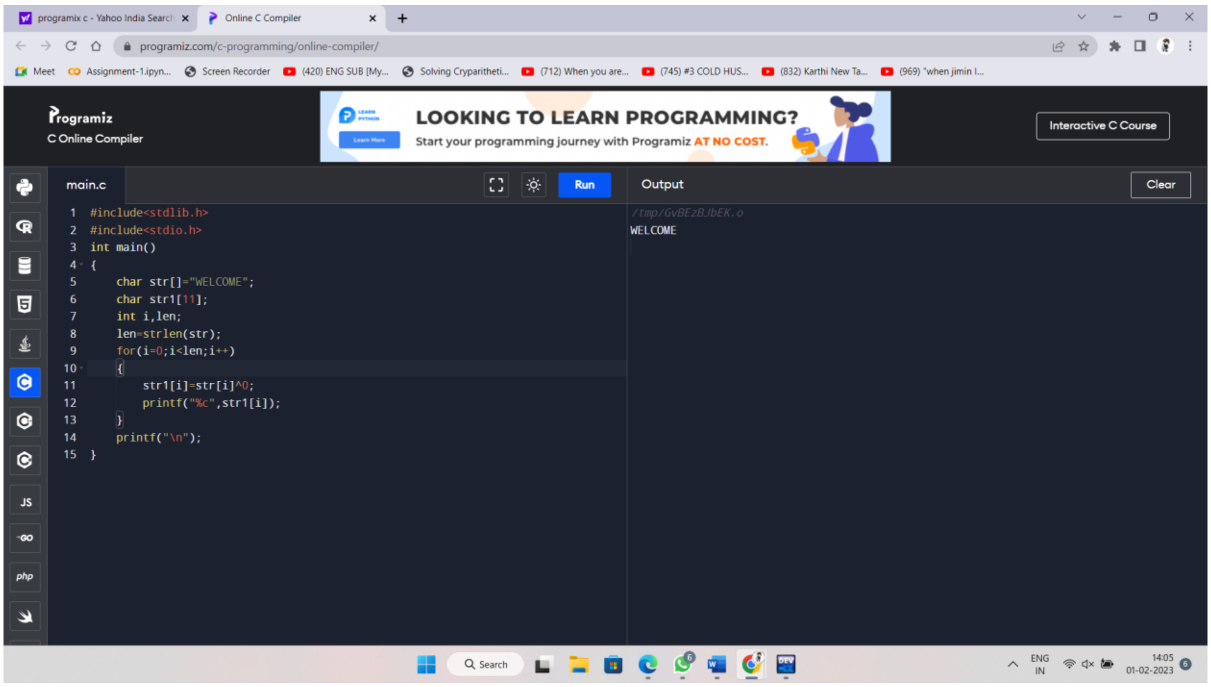
CNS-Programs

R. Aishwarya

192125001

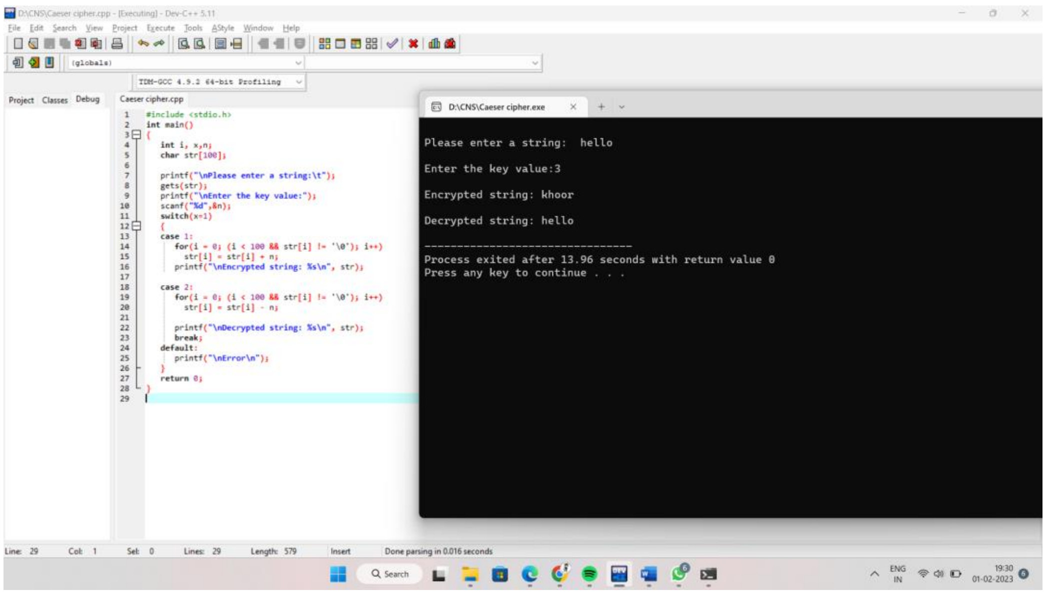
1.Print the elements using the XOR.



```c
#include<stdlib.h>
#include<stdio.h>
int main()
{
    char str[]="WELCOME";
    char str1[11];
    int i,len;
    len=strlen(str);
    for(i=0;i<len;i++)
    {
        str1[i]=str[i]^0;
        printf("%c",str1[i]);
    }
    printf("\n");
}
```

```
/tmp/GvBEzBJbEK.o
WELCOME
```

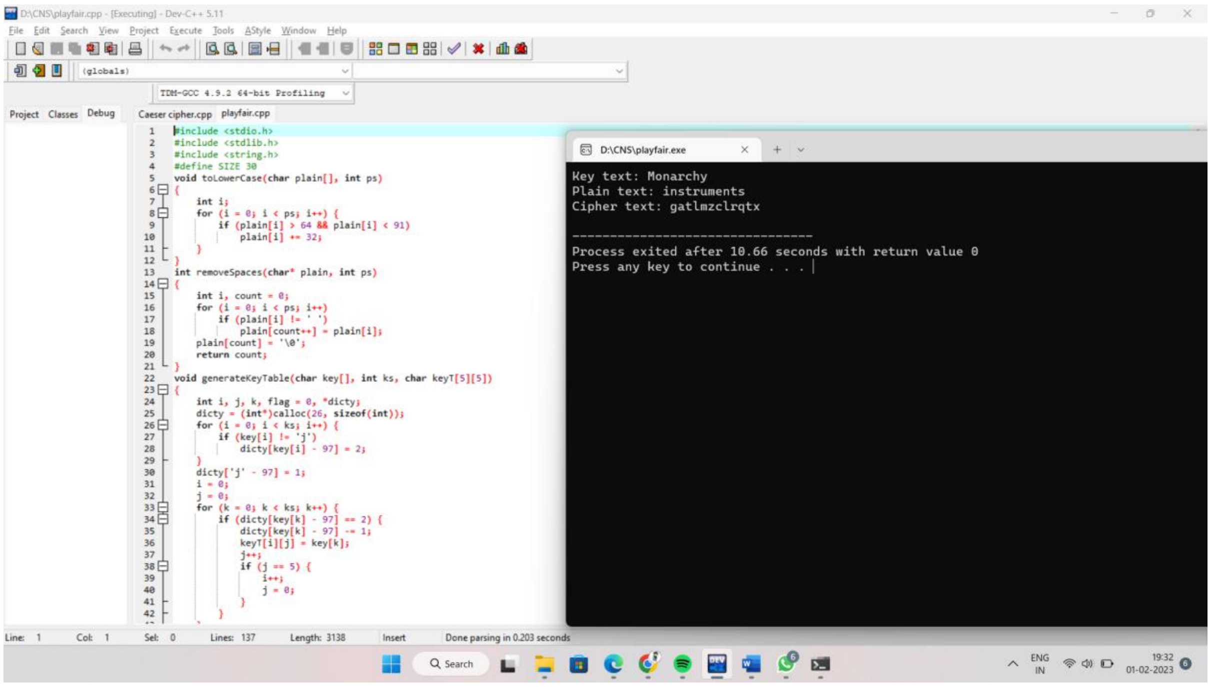
2. Write a C program for Caesar cipher involves replacing each letter of the alphabet with the letter stan places further down the alphabet, for k in the range 1 through 25.



```c
#include <stdio.h>
int main()
{
    int i, x,n;
    char str[100];

    printf("\nPlease enter a string:\t");
    gets(str);
    printf("\nEnter the key value:");
    scanf("%d",&n);
    switch(x=1)
    {
    case 1:
        for(i = 0; (i < 100 && str[i] != '\0'); i++)
          str[i] = str[i] + n;
        printf("\nEncrypted string: %s\n", str);

    case 2:
        for(i = 0; (i < 100 && str[i] != '\0'); i++)
          str[i] = str[i] - n;

        printf("\nDecrypted string: %s\n", str);
        break;
    default:
        printf("\nError\n");
    }
    return 0;
}
```

```
Please enter a string:  hello

Enter the key value:3

Encrypted string: khoor

Decrypted string: hello

--------------------------------
Process exited after 13.96 seconds with return value 0
Press any key to continue . . .
```

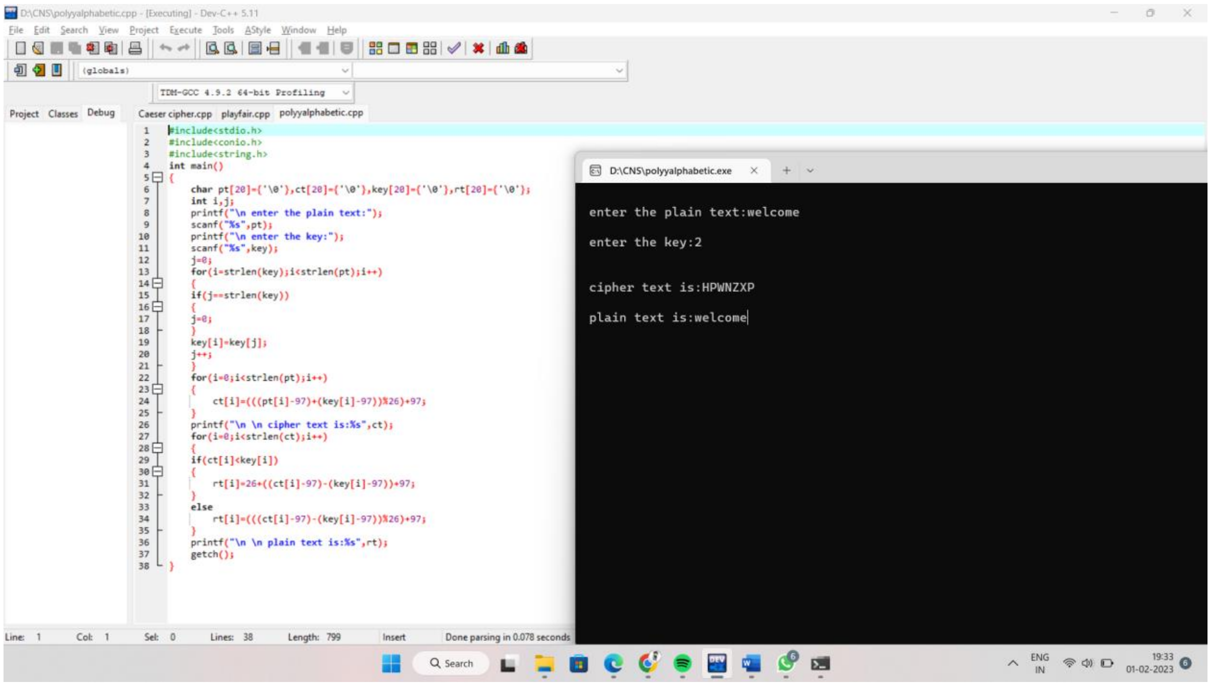
3. . Write a C program for Playfair algorithm is based on the use of a 5 X 5 matrix of letters constructed using a keyword. Plaintext is encrypted two letters at a time using this matrix.



```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define SIZE 30
void toLowerCase(char plain[], int ps)
{
    int i;
    for (i = 0; i < ps; i++) {
        if (plain[i] > 64 && plain[i] < 91)
            plain[i] += 32;
    }
}
int removeSpaces(char* plain, int ps)
{
    int i, count = 0;
    for (i = 0; i < ps; i++)
        if (plain[i] != ' ')
            plain[count++] = plain[i];
    plain[count] = '\0';
    return count;
}
void generateKeyTable(char key[], int ks, char keyT[5][5])
{
    int i, j, k, flag = 0, *dicty;
    dicty = (int*)calloc(26, sizeof(int));
    for (i = 0; i < ks; i++) {
        if (key[i] != 'j')
            dicty[key[i] - 97] = 2;
    }
    dicty['j' - 97] = 1;
    i = 0;
    j = 0;
    for (k = 0; k < ks; k++) {
        if (dicty[key[k] - 97] == 2) {
            dicty[key[k] - 97] -= 1;
            keyT[i][j] = key[k];
            j++;
            if (j == 5) {
                i++;
                j = 0;
            }
        }
```

```
Key text: Monarchy
Plain text: instruments
Cipher text: gatlmzclrqtx

--------------------------------
Process exited after 10.66 seconds with return value 0
Press any key to continue . . .
```

4. Write a C program for polyalphabetic substitution cipher uses a separate monoalphabetic substitution cipher for each successive letter of plaintext, depending on a key.



```c
#include<stdio.h>
#include<conio.h>
#include<string.h>
int main()
{
    char pt[20]={'\0'},ct[20]={'\0'},key[20]={'\0'},rt[20]={'\0'};
    int i,j;
    printf("\n enter the plain text:");
    scanf("%s",pt);
    printf("\n enter the key:");
    scanf("%s",key);
    j=0;
    for(i=strlen(key);i<strlen(pt);i++)
    {
    if(j==strlen(key))
    {
    j=0;
    }
    key[i]=key[j];
    j++;
    }
    for(i=0;i<strlen(pt);i++)
    {
        ct[i]=(((pt[i]-97)+(key[i]-97))%26)+97;
    }
    printf("\n \n cipher text is:%s",ct);
    for(i=0;i<strlen(ct);i++)
    {
    if(ct[i]<key[i])
    {
        rt[i]=26+((ct[i]-97)-(key[i]-97))+97;
    }
    else
        rt[i]=(((ct[i]-97)-(key[i]-97))%26)+97;
    }
    printf("\n \n plain text is:%s",rt);
    getch();
}
```

```
enter the plain text:welcome

enter the key:2


cipher text is:HPWNZXP

plain text is:welcome
```