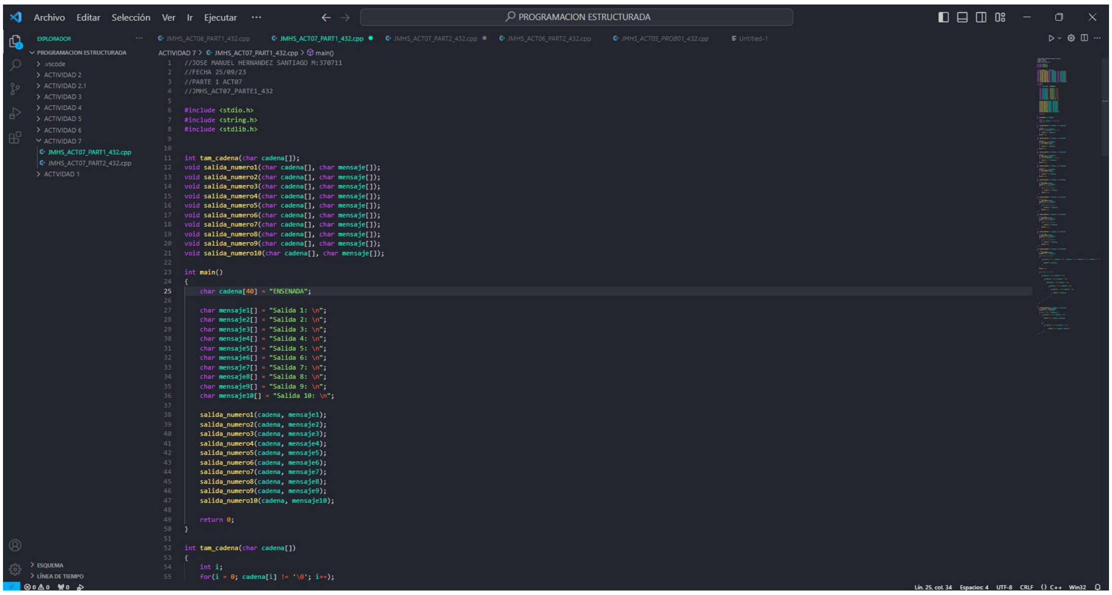
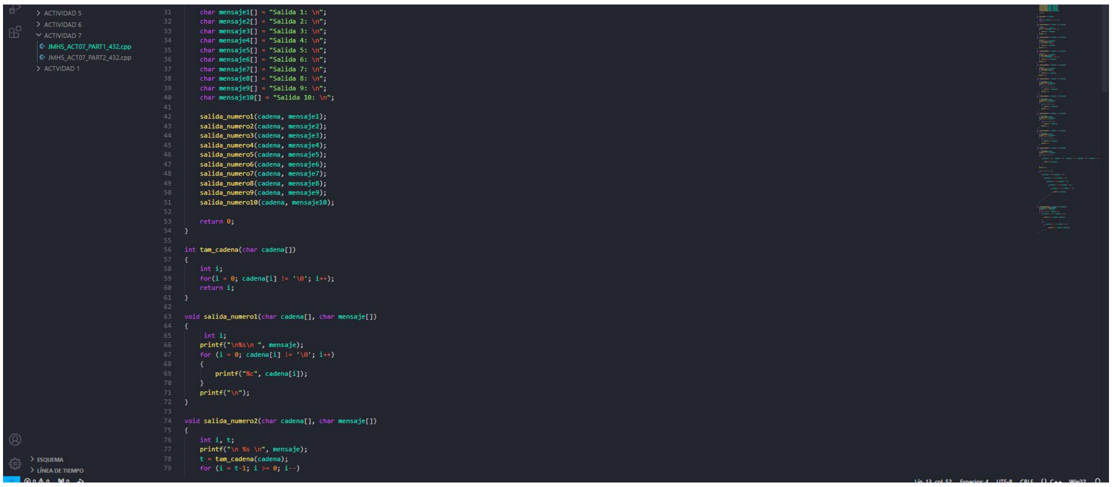
ANEXOS
ACTIVIDAD 7
PERTE 1 Y 2
JOSE MANUEL HERNANDEZ SANTIAGO M:370711
PROGRAMAION ESTRUCTURADA
GRUPO:432

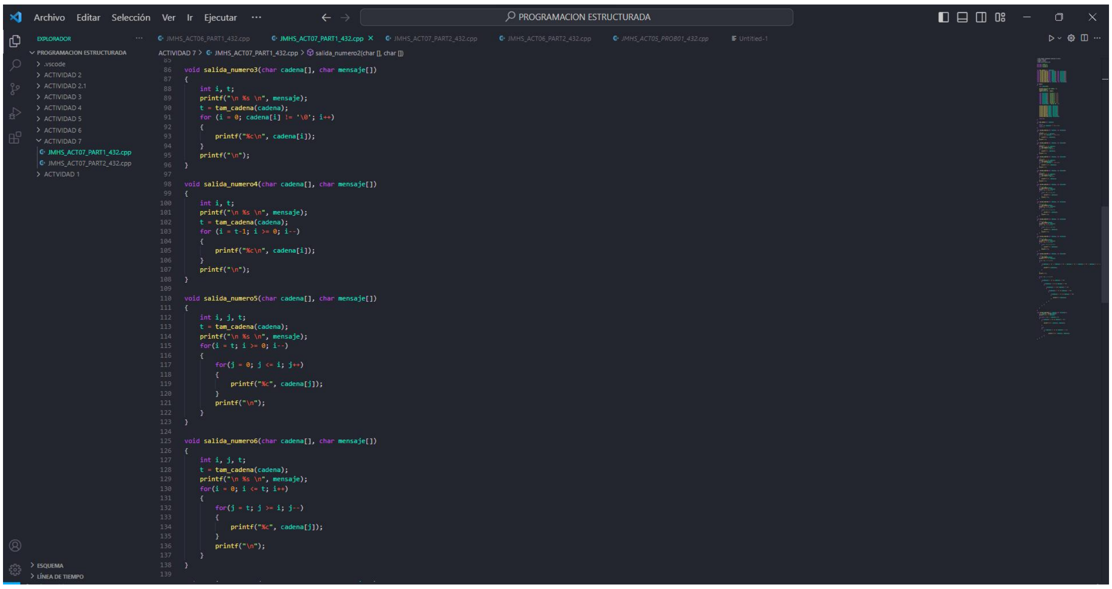
PARTE 1:



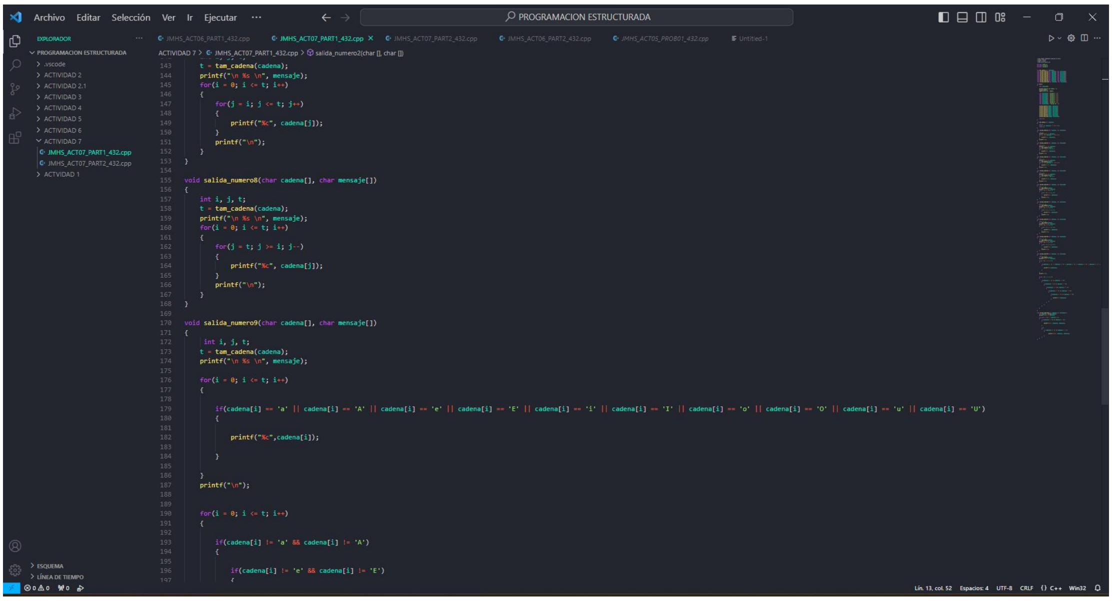
```c
//JOSE MANUEL HERNANDEZ SANTIAGO M:370711
//FECHA 25/09/23
//PARTE 1 ACT07
//JMHS_ACT07_PARTE1_432

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


int tam_cadena(char cadena[]);
void salida_numero1(char cadena[], char mensaje[]);
void salida_numero2(char cadena[], char mensaje[]);
void salida_numero3(char cadena[], char mensaje[]);
void salida_numero4(char cadena[], char mensaje[]);
void salida_numero5(char cadena[], char mensaje[]);
void salida_numero6(char cadena[], char mensaje[]);
void salida_numero7(char cadena[], char mensaje[]);
void salida_numero8(char cadena[], char mensaje[]);
void salida_numero9(char cadena[], char mensaje[]);
void salida_numero10(char cadena[], char mensaje[]);

int main()
{
    char cadena[40] = "ENSENADA";

    char mensaje1[] = "Salida 1: \n";
    char mensaje2[] = "Salida 2: \n";
    char mensaje3[] = "Salida 3: \n";
    char mensaje4[] = "Salida 4: \n";
    char mensaje5[] = "Salida 5: \n";
    char mensaje6[] = "Salida 6: \n";
    char mensaje7[] = "Salida 7: \n";
    char mensaje8[] = "Salida 8: \n";
    char mensaje9[] = "Salida 9: \n";
    char mensaje10[] = "Salida 10: \n";

    salida_numero1(cadena, mensaje1);
    salida_numero2(cadena, mensaje2);
    salida_numero3(cadena, mensaje3);
    salida_numero4(cadena, mensaje4);
    salida_numero5(cadena, mensaje5);
    salida_numero6(cadena, mensaje6);
    salida_numero7(cadena, mensaje7);
    salida_numero8(cadena, mensaje8);
    salida_numero9(cadena, mensaje9);
    salida_numero10(cadena, mensaje10);

    return 0;
}

int tam_cadena(char cadena[])
{
    int i;
    for(i = 0; cadena[i] != '\0'; i++);
```



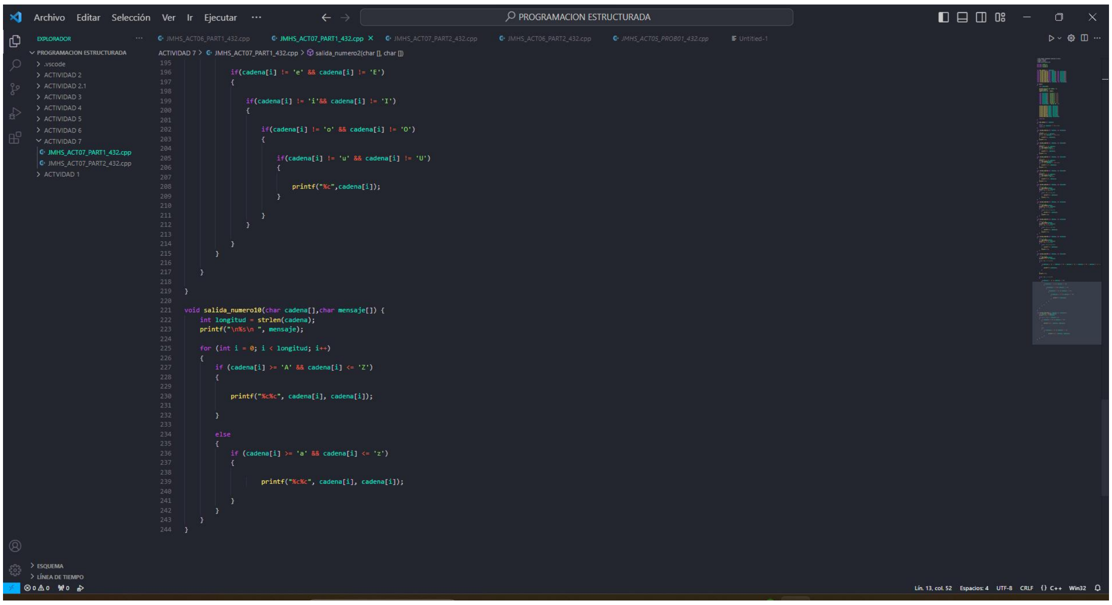
```c
    char mensaje1[] = "Salida 1: \n";
    char mensaje2[] = "Salida 2: \n";
    char mensaje3[] = "Salida 3: \n";
    char mensaje4[] = "Salida 4: \n";
    char mensaje5[] = "Salida 5: \n";
    char mensaje6[] = "Salida 6: \n";
    char mensaje7[] = "Salida 7: \n";
    char mensaje8[] = "Salida 8: \n";
    char mensaje9[] = "Salida 9: \n";
    char mensaje10[] = "Salida 10: \n";

    salida_numero1(cadena, mensaje1);
    salida_numero2(cadena, mensaje2);
    salida_numero3(cadena, mensaje3);
    salida_numero4(cadena, mensaje4);
    salida_numero5(cadena, mensaje5);
    salida_numero6(cadena, mensaje6);
    salida_numero7(cadena, mensaje7);
    salida_numero8(cadena, mensaje8);
    salida_numero9(cadena, mensaje9);
    salida_numero10(cadena, mensaje10);

    return 0;
}

int tam_cadena(char cadena[])
{
    int i;
    for(i = 0; cadena[i] != '\0'; i++);
    return i;
}

void salida_numero1(char cadena[], char mensaje[])
{
     int i;
    printf("\n%s\n ", mensaje);
    for (i = 0; cadena[i] != '\0'; i++)
    {
        printf("%c", cadena[i]);
    }
    printf("\n");
}

void salida_numero2(char cadena[], char mensaje[])
{
    int i, t;
    printf("\n %s \n", mensaje);
    t = tam_cadena(cadena);
    for (i = t-1; i >= 0; i--)
```



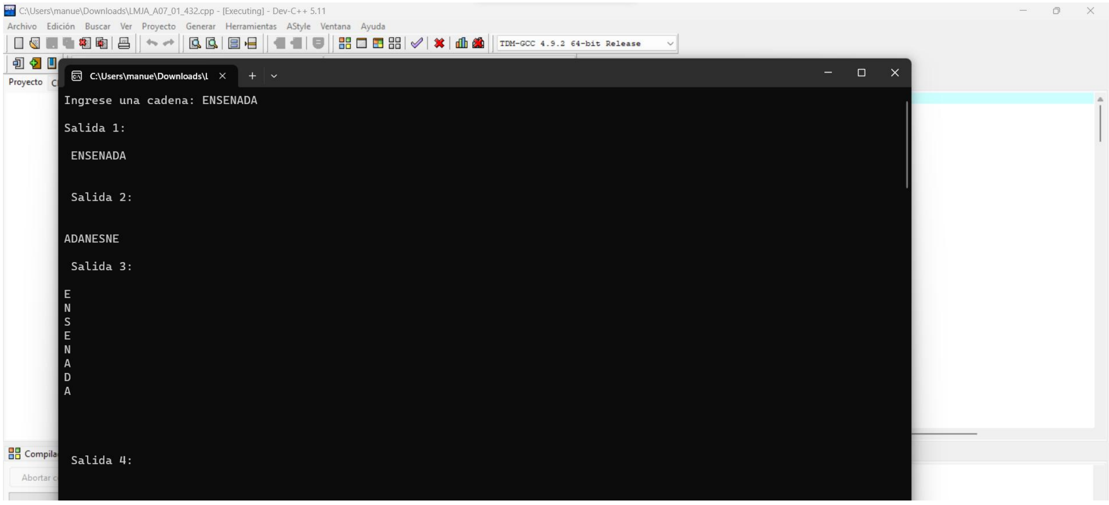
```
void salida_numero3(char cadena[], char mensaje[])
{
    int i, t;
    printf("\n %s \n", mensaje);
    t = tam_cadena(cadena);
    for (i = 0; cadena[i] != '\0'; i++)
    {
        printf("%c\n", cadena[i]);
    }
    printf("\n");
}

void salida_numero4(char cadena[], char mensaje[])
{
    int i, t;
    printf("\n %s \n", mensaje);
    t = tam_cadena(cadena);
    for (i = t-1; i >= 0; i--)
    {
        printf("%c\n", cadena[i]);
    }
    printf("\n");
}

void salida_numero5(char cadena[], char mensaje[])
{
    int i, j, t;
    t = tam_cadena(cadena);
    printf("\n %s \n", mensaje);
    for(i = t; i >= 0; i--)
    {
        for(j = 0; j <= i; j++)
        {
            printf("%c", cadena[j]);
        }
        printf("\n");
    }
}

void salida_numero6(char cadena[], char mensaje[])
{
    int i, j, t;
    t = tam_cadena(cadena);
    printf("\n %s \n", mensaje);
    for(i = 0; i <= t; i++)
    {
        for(j = t; j >= i; j--)
        {
            printf("%c", cadena[j]);
        }
        printf("\n");
    }
}

```


Archivo
Editar
Selección
Ver
Ir
Ejecutar
PROGRAMACION ESTRUCTURADA
EXPLORADOR
ACTIVIDAD 7 > JMHS_ACT07_PART1_432.cpp > salida_numero2(char [], char [])
Lín. 13, col. 52
Espacios: 4
UTF-8
CRLF
{} C++
Win32

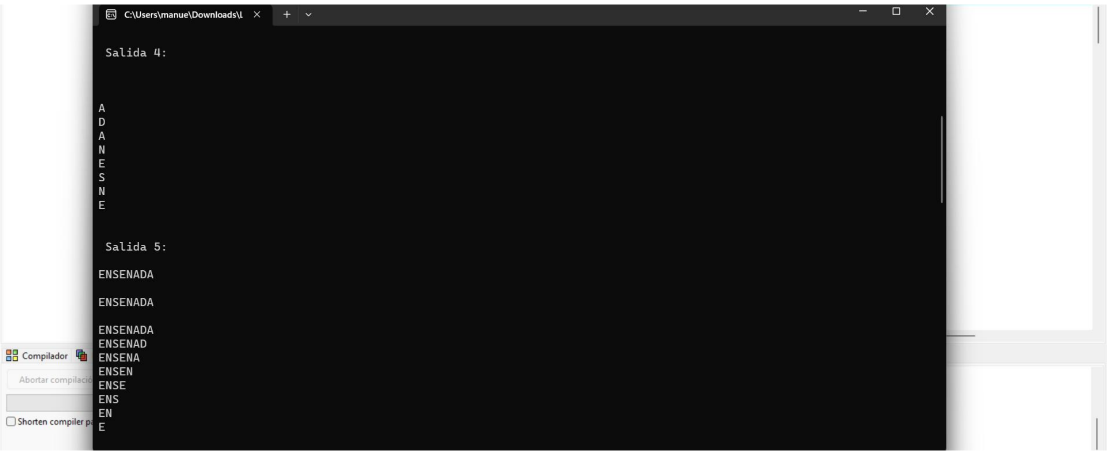
```
    t = tam_cadena(cadena);
    printf("\n %s \n", mensaje);
    for(i = 0; i <= t; i++)
    {
        for(j = i; j <= t; j++)
        {
            printf("%c", cadena[j]);
        }
        printf("\n");
    }
}

void salida_numero8(char cadena[], char mensaje[])
{
    int i, j, t;
    t = tam_cadena(cadena);
    printf("\n %s \n", mensaje);
    for(i = 0; i <= t; i++)
    {
        for(j = t; j >= i; j--)
        {
            printf("%c", cadena[j]);
        }
        printf("\n");
    }
}

void salida_numero9(char cadena[], char mensaje[])
{
     int i, j, t;
    t = tam_cadena(cadena);
    printf("\n %s \n", mensaje);

    for(i = 0; i <= t; i++)
    {

        if(cadena[i] == 'a' || cadena[i] == 'A' || cadena[i] == 'e' || cadena[i] == 'E' || cadena[i] == 'i' || cadena[i] == 'I' || cadena[i] == 'o' || cadena[i] == 'O' || cadena[i] == 'u' || cadena[i] == 'U')
        {

            printf("%c",cadena[i]);

        }

    }
    printf("\n");


    for(i = 0; i <= t; i++)
    {

        if(cadena[i] != 'a' && cadena[i] != 'A')
        {

            if(cadena[i] != 'e' && cadena[i] != 'E')
```


Archivo
Editar
Selección
Ver
Ir
Ejecutar
PROGRAMACION ESTRUCTURADA
EXPLORADOR
ACTIVIDAD 7 > JMHS_ACT07_PART1_432.cpp > salida_numero2(char [], char [])
Lín. 13, col. 52
Espacios: 4
UTF-8
CRLF
{} C++
Win32

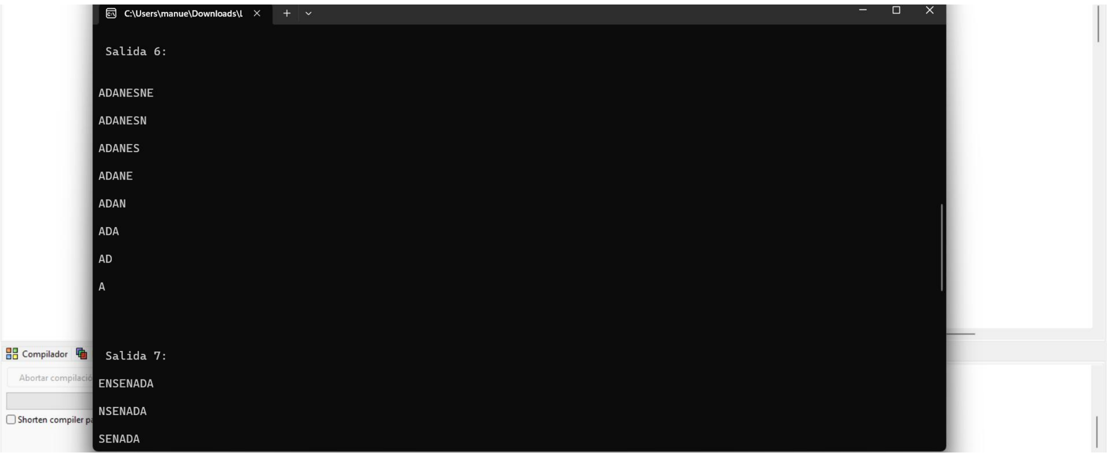
```

            if(cadena[i] != 'e' && cadena[i] != 'E')
            {

                if(cadena[i] != 'i'&& cadena[i] != 'I')
                {

                    if(cadena[i] != 'o' && cadena[i] != 'O')
                    {

                        if(cadena[i] != 'u' && cadena[i] != 'U')
                        {

                            printf("%c",cadena[i]);
                        }

                    }
                }

            }
        }

    }

}

void salida_numero10(char cadena[],char mensaje[]) {
    int longitud = strlen(cadena);
    printf("\n%s\n ", mensaje);

    for (int i = 0; i < longitud; i++)
    {
        if (cadena[i] >= 'A' && cadena[i] <= 'Z')
        {

            printf("%c%c", cadena[i], cadena[i]);

        }

        else
        {
            if (cadena[i] >= 'a' && cadena[i] <= 'z')
            {

                printf("%c%c", cadena[i], cadena[i]);

            }
        }
    }
}
```


C:\Users\manue\Downloads\LMJA_A07_01_432.cpp - [Executing] - Dev-C++ 5.11
Archivo Edición Buscar Ver Proyecto Generar Herramientas AStyle Ventana Ayuda
TDM-GCC 4.9.2 64-bit Release
Proyecto
C:\Users\manue\Downloads\L
Compila
Abortar c

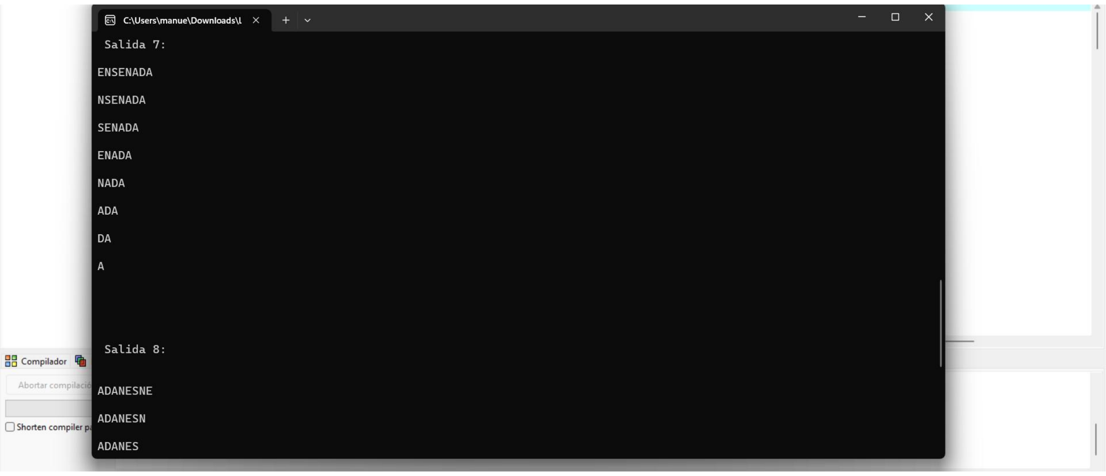
```
Ingrese una cadena: ENSENADA

Salida 1:

 ENSENADA


 Salida 2:


ADANESNE

 Salida 3:

E
N
S
E
N
A
D
A




 Salida 4:
```



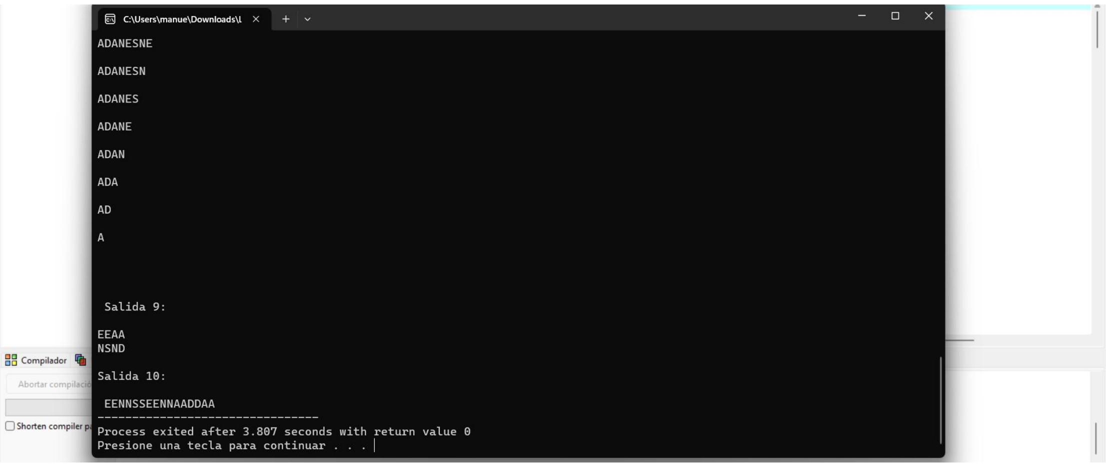
```
 Salida 4:



A
D
A
N
E
S
N
E


 Salida 5:

ENSENADA

ENSENADA

ENSENADA
ENSENAD
ENSENA
ENSEN
ENSE
ENS
EN
E
```



```
 Salida 6:


ADANESNE

ADANESN

ADANES

ADANE

ADAN

ADA

AD

A



 Salida 7:
ENSENADA

NSENADA

SENADA
```

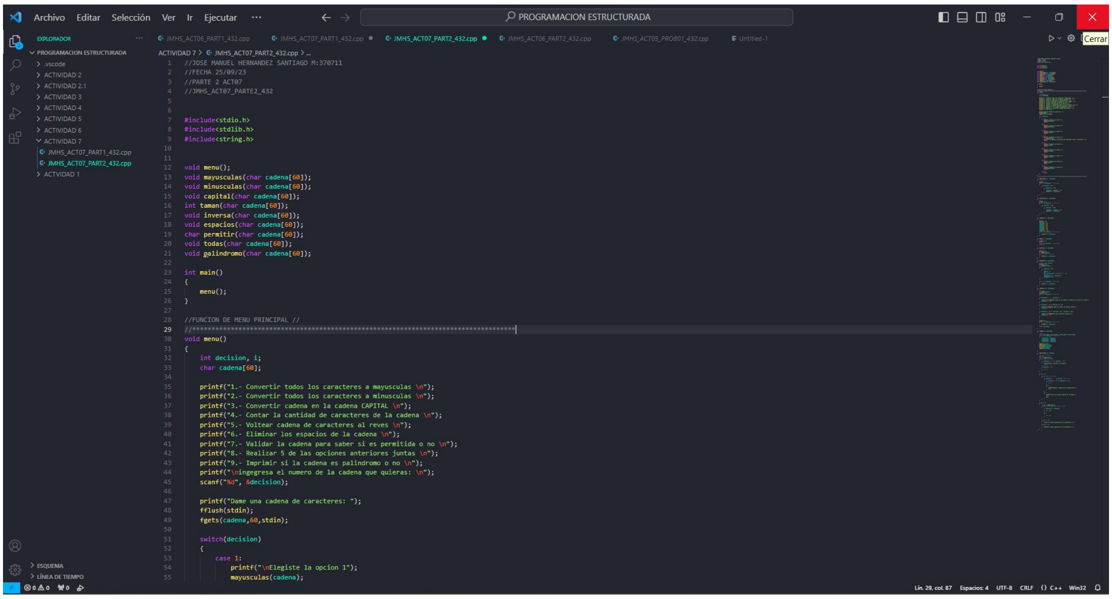


```
 Salida 7:

ENSENADA

NSENADA

SENADA

ENADA

NADA

ADA

DA

A



 Salida 8:

ADANESNE

ADANESN

ADANES
```

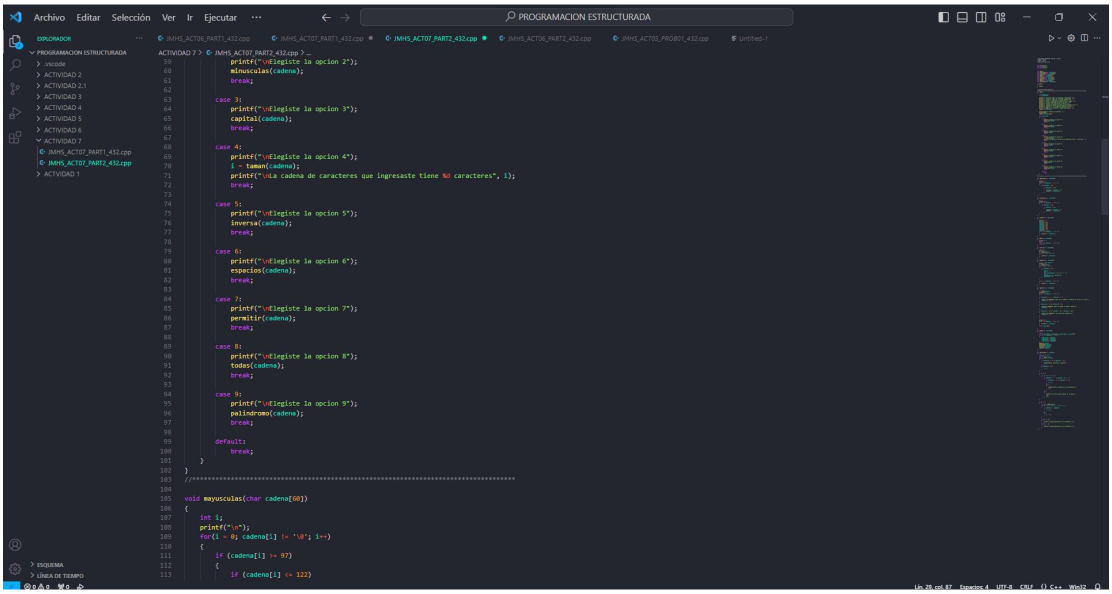


```
ADANESNE

ADANESN

ADANES

ADANE

ADAN

ADA

AD

A



 Salida 9:

EEAA
NSND

Salida 10:

 EENNSSEENNAADDAA
--------------------------------
Process exited after 3.807 seconds with return value 0
Presione una tecla para continuar . . .
```

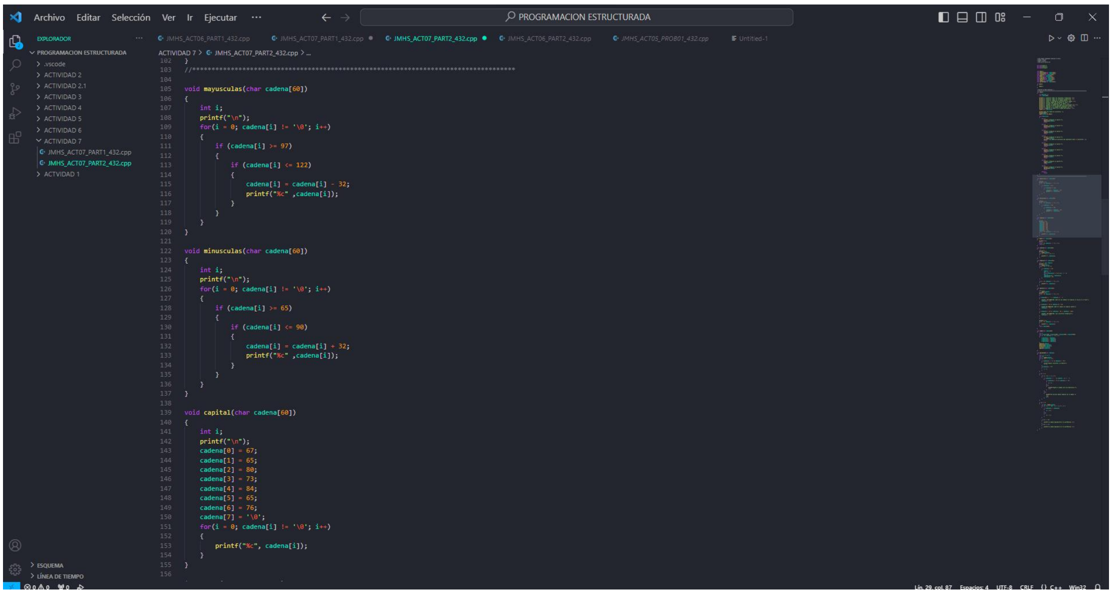
PARTE 2:



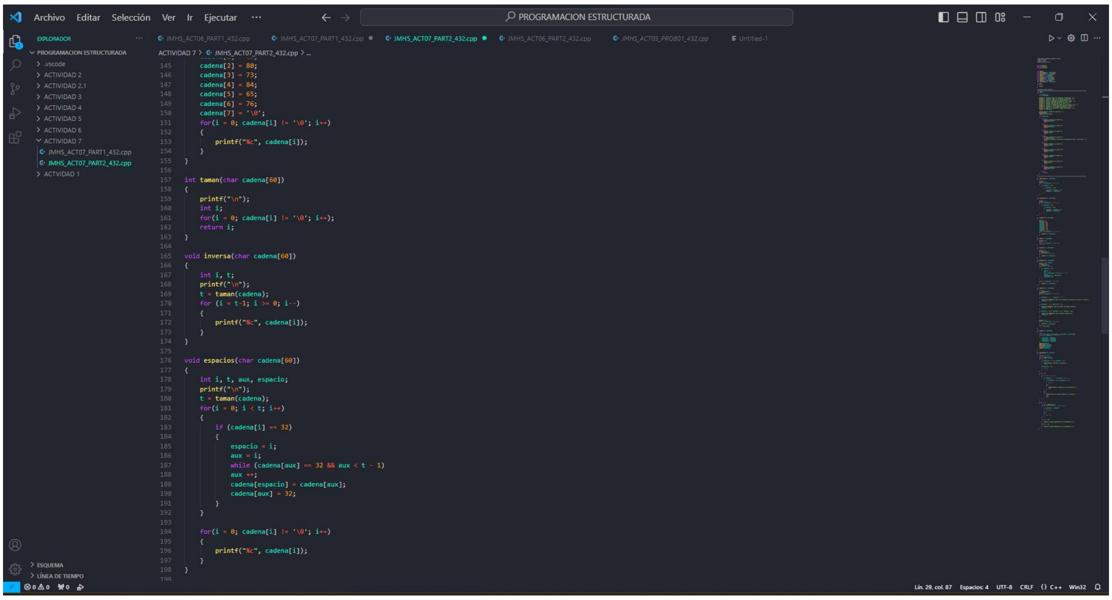
```c
//JOSE MANUEL HERNANDEZ SANTIAGO M:370711
//FECHA 25/09/23
//PARTE 2 ACT07
//JMHS_ACT07_PARTE2_432


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


void menu();
void mayusculas(char cadena[60]);
void minusculas(char cadena[60]);
void capital(char cadena[60]);
int taman(char cadena[60]);
void inversa(char cadena[60]);
void espacios(char cadena[60]);
char permitir(char cadena[60]);
void todas(char cadena[60]);
void palindromo(char cadena[60]);

int main()
{
    menu();
}

//FUNCION DE MENU PRINCIPAL //
//**********************************************************************************
void menu()
{
    int decision, i;
    char cadena[60];

    printf("1.- Convertir todos los caracteres a mayusculas \n");
    printf("2.- Convertir todos los caracteres a minusculas \n");
    printf("3.- Convertir cadena en la cadena CAPITAL \n");
    printf("4.- Contar la cantidad de caracteres de la cadena \n");
    printf("5.- Voltear cadena de caracteres al reves \n");
    printf("6.- Eliminar los espacios de la cadena \n");
    printf("7.- Validar la cadena para saber si es permitida o no \n");
    printf("8.- Realizar 5 de las opciones anteriores juntas \n");
    printf("9.- Imprimir si la cadena es palindromo o no \n");
    printf("\ningegresa el numero de la cadena que quieras: \n");
    scanf("%d", &decision);

    printf("Dame una cadena de caracteres: ");
    fflush(stdin);
    fgets(cadena,60,stdin);

    switch(decision)
    {
        case 1:
            printf("\nElegiste la opcion 1");
            mayusculas(cadena);
```



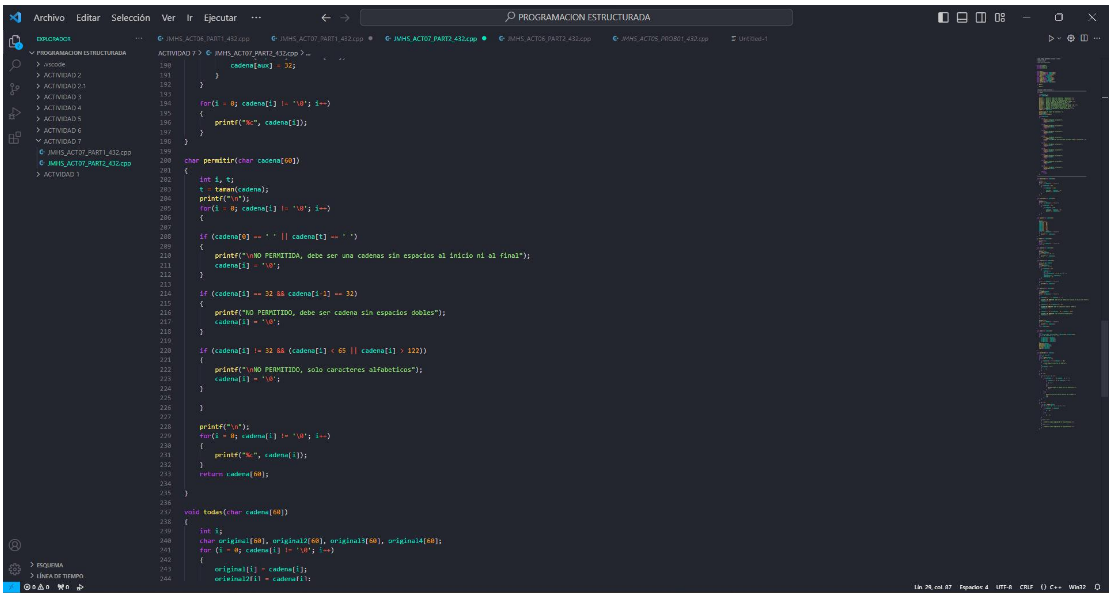
```
            printf("\nElegiste la opcion 2");
            minusculas(cadena);
            break;

        case 3:
            printf("\nElegiste la opcion 3");
            capital(cadena);
            break;

        case 4:
            printf("\nElegiste la opcion 4");
            i = taman(cadena);
            printf("\nLa cadena de caracteres que ingresaste tiene %d caracteres", i);
            break;

        case 5:
            printf("\nElegiste la opcion 5");
            inversa(cadena);
            break;

        case 6:
            printf("\nElegiste la opcion 6");
            espacios(cadena);
            break;

        case 7:
            printf("\nElegiste la opcion 7");
            permitir(cadena);
            break;

        case 8:
            printf("\nElegiste la opcion 8");
            todas(cadena);
            break;

        case 9:
            printf("\nElegiste la opcion 9");
            palindromo(cadena);
            break;

        default:
            break;
    }
}
//************************************************************************************

void mayusculas(char cadena[60])
{
    int i;
    printf("\n");
    for(i = 0; cadena[i] != '\0'; i++)
    {
        if (cadena[i] >= 97)
        {
            if (cadena[i] <= 122)
```



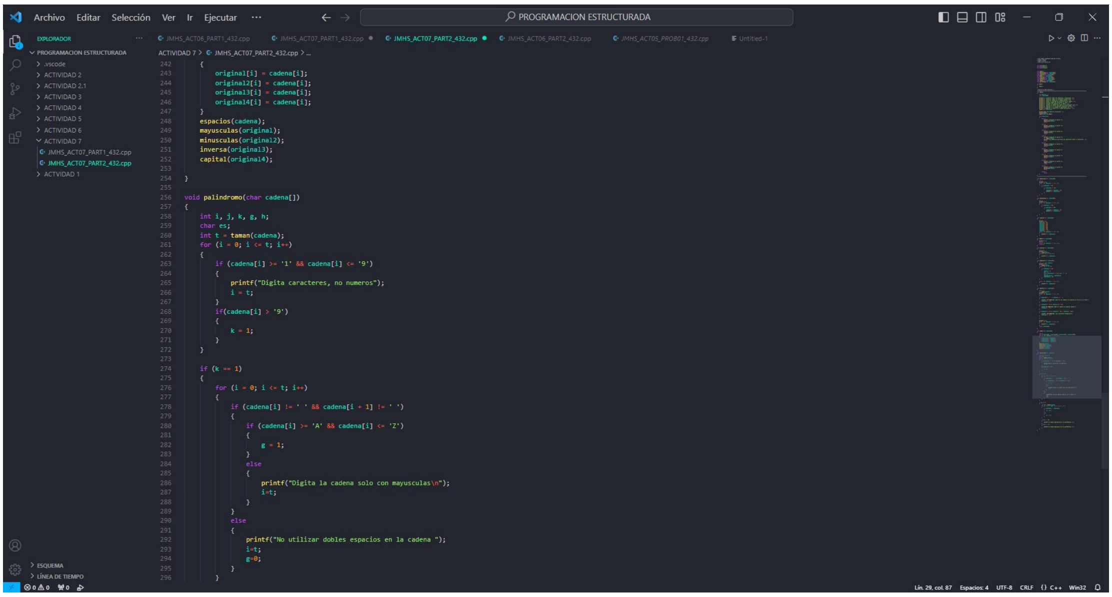
```
}
//************************************************************************************

void mayusculas(char cadena[60])
{
    int i;
    printf("\n");
    for(i = 0; cadena[i] != '\0'; i++)
    {
        if (cadena[i] >= 97)
        {
            if (cadena[i] <= 122)
            {
                cadena[i] = cadena[i] - 32;
                printf("%c" ,cadena[i]);
            }
        }
    }
}

void minusculas(char cadena[60])
{
    int i;
    printf("\n");
    for(i = 0; cadena[i] != '\0'; i++)
    {
        if (cadena[i] >= 65)
        {
            if (cadena[i] <= 90)
            {
                cadena[i] = cadena[i] + 32;
                printf("%c" ,cadena[i]);
            }
        }
    }
}

void capital(char cadena[60])
{
    int i;
    printf("\n");
    cadena[0] = 67;
    cadena[1] = 65;
    cadena[2] = 80;
    cadena[3] = 73;
    cadena[4] = 84;
    cadena[5] = 65;
    cadena[6] = 76;
    cadena[7] = '\0';
    for(i = 0; cadena[i] != '\0'; i++)
    {
        printf("%c", cadena[i]);
    }
}

```



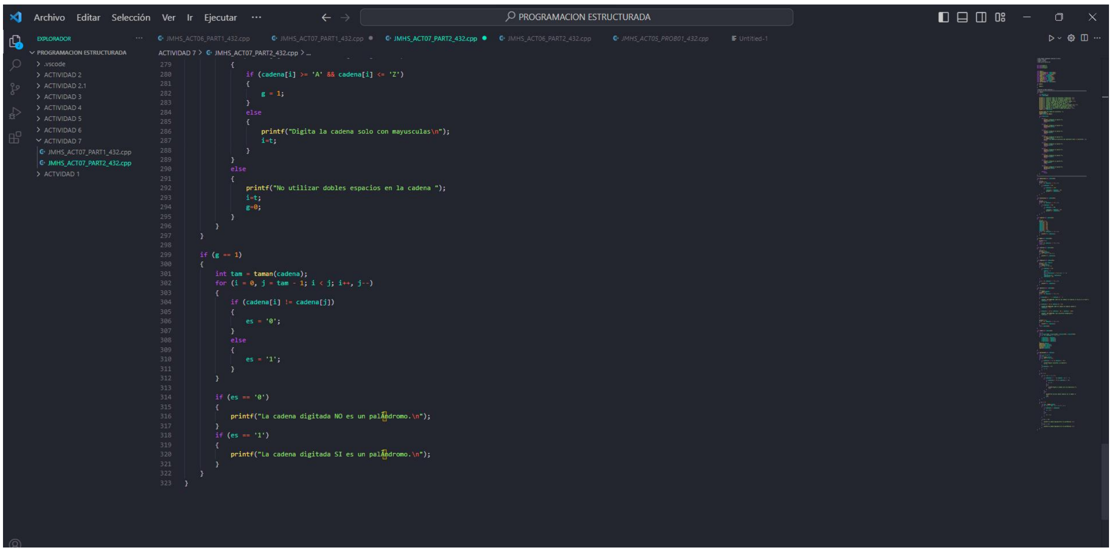
```
    cadena[2] = 80;
    cadena[3] = 73;
    cadena[4] = 84;
    cadena[5] = 65;
    cadena[6] = 76;
    cadena[7] = '\0';
    for(i = 0; cadena[i] != '\0'; i++)
    {
        printf("%c", cadena[i]);
    }
}

int taman(char cadena[60])
{
    printf("\n");
    int i;
    for(i = 0; cadena[i] != '\0'; i++);
    return i;
}

void inversa(char cadena[60])
{
    int i, t;
    printf("\n");
    t = taman(cadena);
    for (i = t-1; i >= 0; i--)
    {
        printf("%c", cadena[i]);
    }
}

void espacios(char cadena[60])
{
    int i, t, aux, espacio;
    printf("\n");
    t = taman(cadena);
    for(i = 0; i < t; i++)
    {
        if (cadena[i] == 32)
        {
            espacio = i;
            aux = i;
            while (cadena[aux] == 32 && aux < t - 1)
            aux ++;
            cadena[espacio] = cadena[aux];
            cadena[aux] = 32;
        }
    }

    for(i = 0; cadena[i] != '\0'; i++)
    {
        printf("%c", cadena[i]);
    }
}
```



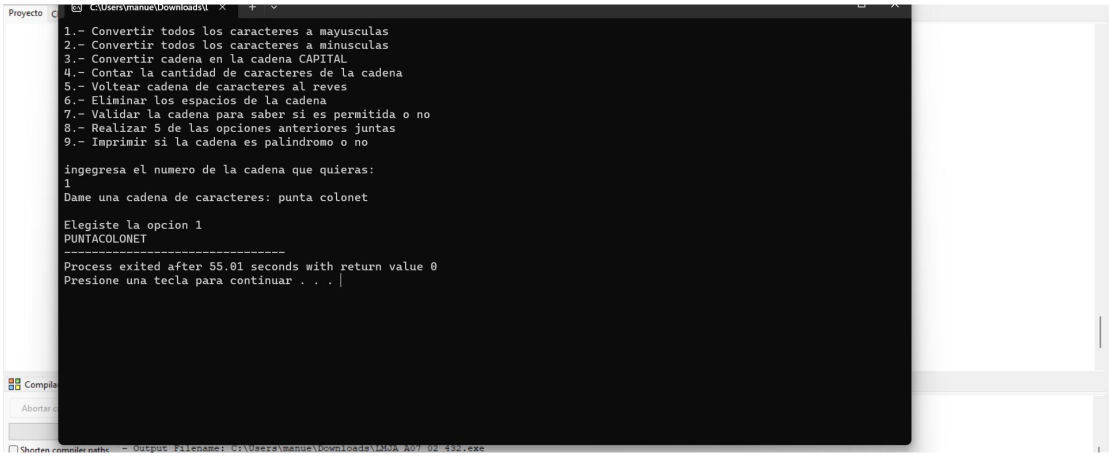
```
                cadena[aux] = 32;
            }
        }

        for(i = 0; cadena[i] != '\0'; i++)
        {
            printf("%c", cadena[i]);
        }
}

char permitir(char cadena[60])
{
    int i, t;
    t = taman(cadena);
    printf("\n");
    for(i = 0; cadena[i] != '\0'; i++)
    {

    if (cadena[0] == ' ' || cadena[t] == ' ')
    {
        printf("\nNO PERMITIDA, debe ser una cadenas sin espacios al inicio ni al final");
        cadena[i] = '\0';
    }

    if (cadena[i] == 32 && cadena[i-1] == 32)
    {
        printf("NO PERMITIDO, debe ser cadena sin espacios dobles");
        cadena[i] = '\0';
    }

    if (cadena[i] != 32 && (cadena[i] < 65 || cadena[i] > 122))
    {
        printf("\nNO PERMITIDO, solo caracteres alfabeticos");
        cadena[i] = '\0';
    }

    }

    printf("\n");
    for(i = 0; cadena[i] != '\0'; i++)
    {
        printf("%c", cadena[i]);
    }
    return cadena[60];

}

void todas(char cadena[60])
{
    int i;
    char original[60], original2[60], original3[60], original4[60];
    for (i = 0; cadena[i] != '\0'; i++)
    {
        original[i] = cadena[i];
        original2[i] = cadena[i];
```


Archivo Editar Selección Ver Ir Ejecutar
PROGRAMACION ESTRUCTURADA
ACTIVIDAD 7 > JMHS_ACT07_PART2_432.cpp > ...
Lín. 29, col. 87 Espacios: 4 UTF-8 CRLF {} C++ Win32

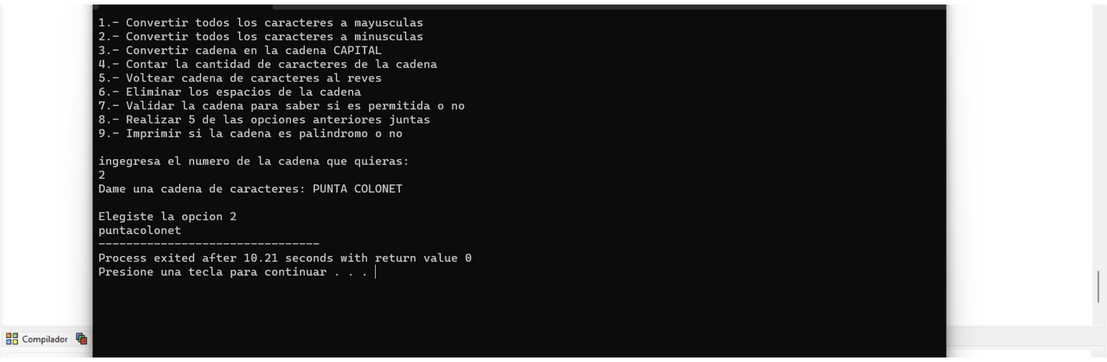
```
    {
        original[i] = cadena[i];
        original2[i] = cadena[i];
        original3[i] = cadena[i];
        original4[i] = cadena[i];
    }
    espacios(cadena);
    mayusculas(original);
    minusculas(original2);
    inversa(original3);
    capital(original4);

}

void palindromo(char cadena[])
{
    int i, j, k, g, h;
    char es;
    int t = taman(cadena);
    for (i = 0; i <= t; i++)
    {
        if (cadena[i] >= '1' && cadena[i] <= '9')
        {
            printf("Digita caracteres, no numeros");
            i = t;
        }
        if(cadena[i] > '9')
        {
            k = 1;
        }
    }

    if (k == 1)
    {
        for (i = 0; i <= t; i++)
        {
            if (cadena[i] != ' ' && cadena[i + 1] != ' ')
            {
                if (cadena[i] >= 'A' && cadena[i] <= 'Z')
                {
                    g = 1;
                }
                else
                {
                    printf("Digita la cadena solo con mayusculas\n");
                    i=t;
                }
            }
            else
            {
                printf("No utilizar dobles espacios en la cadena ");
                i=t;
                g=0;
            }
        }
```


Archivo Editar Selección Ver Ir Ejecutar
PROGRAMACION ESTRUCTURADA
ACTIVIDAD 7 > JMHS_ACT07_PART2_432.cpp > ...

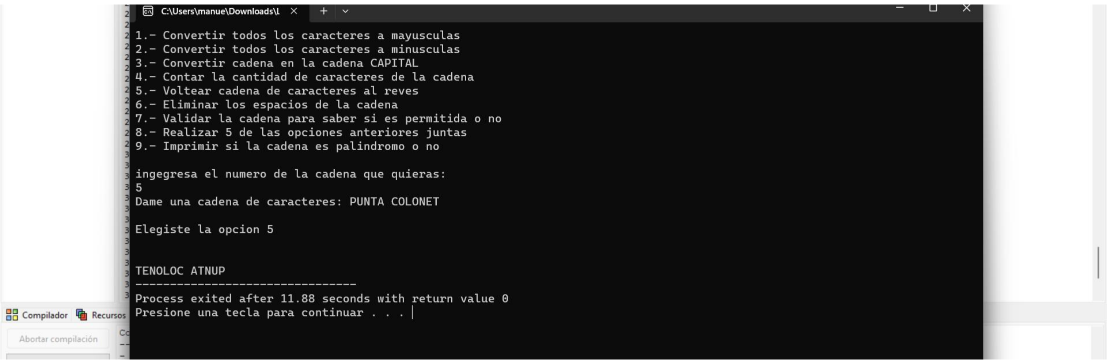
```
            {
                if (cadena[i] >= 'A' && cadena[i] <= 'Z')
                {
                    g = 1;
                }
                else
                {
                    printf("Digita la cadena solo con mayusculas\n");
                    i=t;
                }
            }
            else
            {
                printf("No utilizar dobles espacios en la cadena ");
                i=t;
                g=0;
            }
        }
    }

    if (g == 1)
    {
        int tam = taman(cadena);
        for (i = 0, j = tam - 1; i < j; i++, j--)
        {
            if (cadena[i] != cadena[j])
            {
                es = '0';
            }
            else
            {
                es = '1';
            }
        }

        if (es == '0')
        {
            printf("La cadena digitada NO es un palÁndromo.\n");
        }
        if (es == '1')
        {
            printf("La cadena digitada SI es un palÁndromo.\n");
        }
    }
}
```


Proyecto
C:\Users\manue\Downloads\L
Compila
Abortar c
Shorten compiler paths
- Output Filename: C:\Users\manue\Downloads\LMJA_A07_02_432.exe

```
1.- Convertir todos los caracteres a mayusculas
2.- Convertir todos los caracteres a minusculas
3.- Convertir cadena en la cadena CAPITAL
4.- Contar la cantidad de caracteres de la cadena
5.- Voltear cadena de caracteres al reves
6.- Eliminar los espacios de la cadena
7.- Validar la cadena para saber si es permitida o no
8.- Realizar 5 de las opciones anteriores juntas
9.- Imprimir si la cadena es palindromo o no

ingegresa el numero de la cadena que quieras:
1
Dame una cadena de caracteres: punta colonet

Elegiste la opcion 1
PUNTACOLONET
--------------------------------
Process exited after 55.01 seconds with return value 0
Presione una tecla para continuar . . .
```



```
1.- Convertir todos los caracteres a mayusculas
2.- Convertir todos los caracteres a minusculas
3.- Convertir cadena en la cadena CAPITAL
4.- Contar la cantidad de caracteres de la cadena
5.- Voltear cadena de caracteres al reves
6.- Eliminar los espacios de la cadena
7.- Validar la cadena para saber si es permitida o no
8.- Realizar 5 de las opciones anteriores juntas
9.- Imprimir si la cadena es palindromo o no

ingegresa el numero de la cadena que quieras:
2
Dame una cadena de caracteres: PUNTA COLONET

Elegiste la opcion 2
puntacolonet
--------------------------------
Process exited after 10.21 seconds with return value 0
Presione una tecla para continuar . . .
```



```
1.- Convertir todos los caracteres a mayusculas
2.- Convertir todos los caracteres a minusculas
3.- Convertir cadena en la cadena CAPITAL
4.- Contar la cantidad de caracteres de la cadena
5.- Voltear cadena de caracteres al reves
6.- Eliminar los espacios de la cadena
7.- Validar la cadena para saber si es permitida o no
8.- Realizar 5 de las opciones anteriores juntas
9.- Imprimir si la cadena es palindromo o no

ingegresa el numero de la cadena que quieras:
5
Dame una cadena de caracteres: PUNTA COLONET

Elegiste la opcion 5


TENOLOC ATNUP
--------------------------------
Process exited after 11.88 seconds with return value 0
Presione una tecla para continuar . . .
```