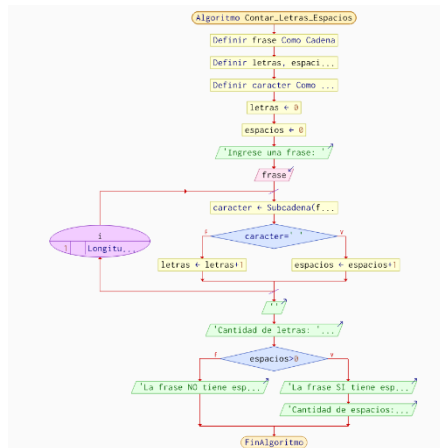


1. El ejercicio debe mostrarnos cuantas letras tiene y si este tiene espacio y cuantos hay en total

```

1 Algoritmo Contar_Letras_Espacios
2 Definir frase Como Cadena
3 Definir letras, espacios, i Como Entero
4 Definir caracter Como Cadena
5
6 letras ← 0
7 espacios ← 0
8
9 Escribir "Ingrese una frase: "
10 Leer frase
11
12 Para i ← 1 Hasta Longitud(frase) Hacer
13     caracter ← Subcadena(frase, i, i)
14     Si caracter = " " Entonces
15         espacios ← espacios + 1
16     Sino
17         letras ← letras + 1
18     FinSi
19 FinPara
20
21 Escribir ""
22 Escribir "Cantidad de letras: ", letras
23
24 Si espacios > 0 Entonces
25     Escribir "La frase SI tiene espacios"
26     Escribir "Cantidad de espacios: ", espacios
27 Sino
28     Escribir "La frase NO tiene espacios"
29 FinSi
30 FinAlgoritmo

```



```

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

int main() {
    char frase[100];
    int letras = 0, espacios = 0;

    printf("Ingrese una frase: ");
    fgets(frase, 100, stdin);

    for (int i = 0; frase[i] != '\0'; i++) {
        if (frase[i] == ' ') {
            espacios++;
        } else if (frase[i] != '\n') {
            letras++;
        }
    }

    printf("\nCantidad de letras: %d\n", letras);

    if (espacios > 0) {
        printf("La frase SI tiene espacios\n");
        printf("Cantidad de espacios: %d\n", espacios);
    } else {
        printf("La frase NO tiene espacios\n");
    }

    return 0;
}

```

<https://onlinegdb.com/XrzG1042A>

## 2. Este ejercicio nos mostrara si la frase entregada es un Palíndromo o no lo

```

Definir texto, limpio, caracter Como Cadena
Definir i, j, n Como Entero
Definir palindromo Como Logico

limpio ← ""
j ← 1
palindromo ← Verdadero

Escribir "Ingrese una frase: "
Leer texto

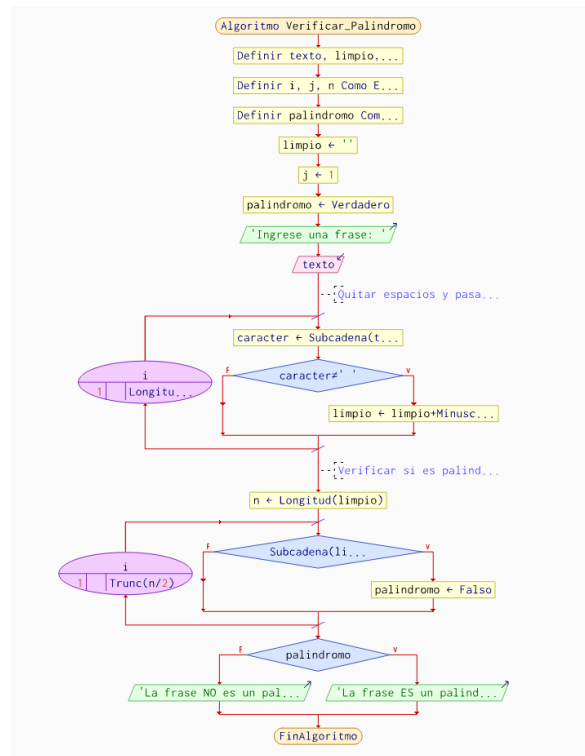
// Quitar espacios y pasar a minusculas
Para i ← 1 Hasta Longitud(texto) Hacer
    caracter ← Subcadena(texto, i, i)
    Si caracter ≠ " " Entonces
        limpio ← limpio + Minusculas(caracter)
    FinSi
FinPara

// Verificar si es palindromo
n ← Longitud(limpio)

Para i ← 1 Hasta Trunc(n / 2) Hacer
    Si Subcadena(limpio, i, i) ≠ Subcadena(limpio, n - i + 1, n - i + 1) Entonces
        palindromo ← Falso
    FinSi
FinPara

Si palindromo Entonces
    Escribir "La frase ES un palindromo"
Sino
    Escribir "La frase NO es un palindromo"
FinSi
FinAlgoritmo

```



es

```

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

int main() {
    char texto[200];
    char limpio[200];
    int i, j = 0;
    int palindromo = 1;

    printf("Ingrese una frase: ");
    fgets(texto, sizeof(texto), stdin);

    /* Quitar salto de línea */
    texto[strcspn(texto, "\n")] = '\0';

    /* Quitar espacios y pasar a minúsculas */
    for (i = 0; texto[i] != '\0'; i++) {
        if (texto[i] != ' ') {
            limpio[j++] = tolower(texto[i]);
        }
    }
    limpio[j] = '\0';

    /* Verificar palíndromo */
    int n = strlen(limpio);
    for (i = 0; i < n / 2; i++) {
        if (limpio[i] != limpio[n - i - 1]) {
            palindromo = 0;
            break;
        }
    }

    if (palindromo)
        printf("La frase ES un palindromo\n");
    else
        printf("La frase NO es un palindromo\n");

    return 0;
}

```

<https://onlinegdb.com/fxxzIVozM>

3. En este ejercicio nos debe unir un nombre y un apellido en una sola cadena con un espacio entre los dos

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

int main() {
    char nombre[100];
    char apellido[100];
    char completo[205];

    printf("Ingrese nombre: ");
    fgets(nombre, sizeof(nombre), stdin);
    nombre[strcspn(nombre, "\n")] = '\0';

    printf("Ingrese apellido: ");
    fgets(apellido, sizeof(apellido), stdin);
    apellido[strcspn(apellido, "\n")] = '\0';

    /* Unir nombre y apellido */
    strcpy(completo, nombre);
    strcat(completo, " ");
    strcat(completo, apellido);

    printf("Nombre completo: %s\n", completo);

    return 0;
}
```

**Algoritmo** Unir\_Nombre\_Apellido

**Definir** nombre, apellido, completo **Como Cadena**

**Escribir** "Ingrese nombre: "

**Leer** nombre

**Escribir** "Ingrese apellido: "

**Leer** apellido

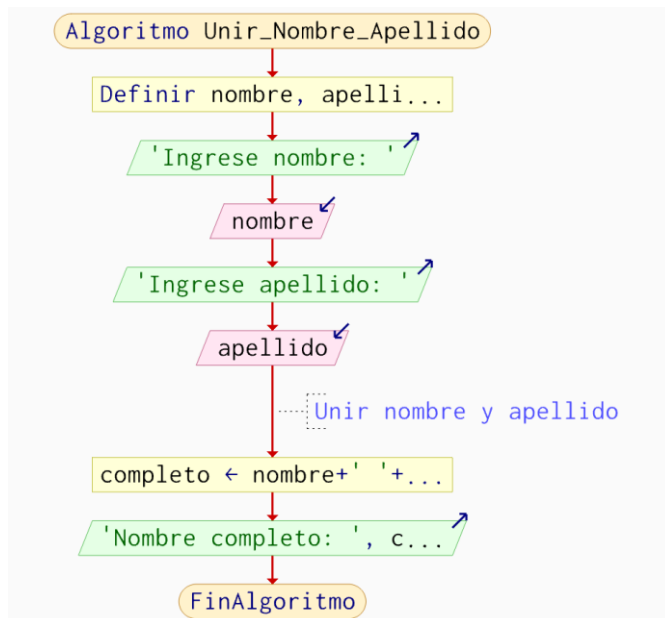
// Unir nombre y apellido

completo ← nombre + " " + apellido

**Escribir** "Nombre completo: ", completo

**FinAlgoritmo**

!;



<https://onlinegdb.com/PjIzC44N4>

4. En este ejercicio buscaremos una palabra dentro de un texto y nos dira cual es la posición de aparición por primera vez

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

int main() {
    char texto[300];
    char palabra[100];
    char *posicion;

    printf("Ingrese un texto: ");
    fgets(texto, sizeof(texto), stdin);
    texto[strlen(texto, "\n")] = '\0';

    printf("Ingrese la palabra a buscar: ");
    fgets(palabra, sizeof(palabra), stdin);
    palabra[strlen(palabra, "\n")] = '\0';

    /* Buscar la palabra dentro del texto */
    posicion = strstr(texto, palabra);

    if (posicion != NULL) {
        int indice = (posicion - texto) + 1; // Posición iniciando desde 1
        printf("La palabra aparece por primera vez en la posición: %d\n", indice);
    } else {
        printf("La palabra no se encontro en el texto\n");
    }

    return 0;
}
```

Algoritmo Buscar\_Palabra\_En\_Texto

Definir texto, palabra Como Cadena  
Definir i, j, pos, lt, lp Como Entero  
Definir encontrado Como Logico

Escribir "Ingrese un texto: "  
Leer texto

Escribir "Ingrese la palabra a buscar: "  
Leer palabra

lt ← Longitud(texto)  
lp ← Longitud(palabra)  
encontrado ← Falso  
pos ← 0

// Buscar la palabra dentro del texto

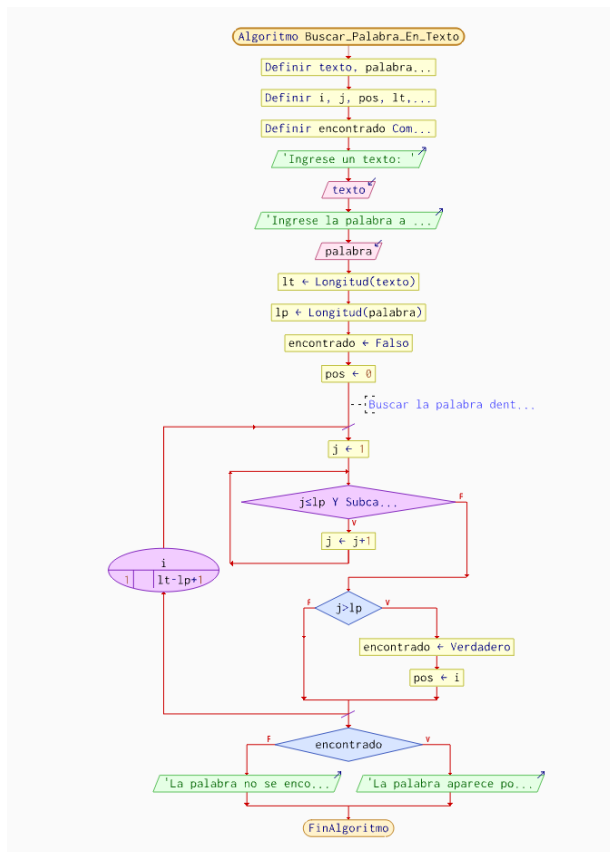
Para i ← 1 Hasta lt - lp + 1 Hacer  
j ← 1

Mientras j ≤ lp Y Subcadena(texto, i + j - 1, i + j - 1) = Subcadena(palabra, j, j) Hacer  
j ← j + 1  
FinMientras

Si j > lp Entonces  
encontrado ← Verdadero  
pos ← i

FinSi  
FinPara

Si encontrado Entonces  
Escribir "La palabra aparece por primera vez en la posición: ", pos  
Sino  
Escribir "La palabra no se encontro en el texto"  
FinSi  
FinAlgoritmo



5. Vamos a convertir una cadena de números a enteros y calcular su cuadrado

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

int main() {
    char numeroCadena[50];
    int numero;

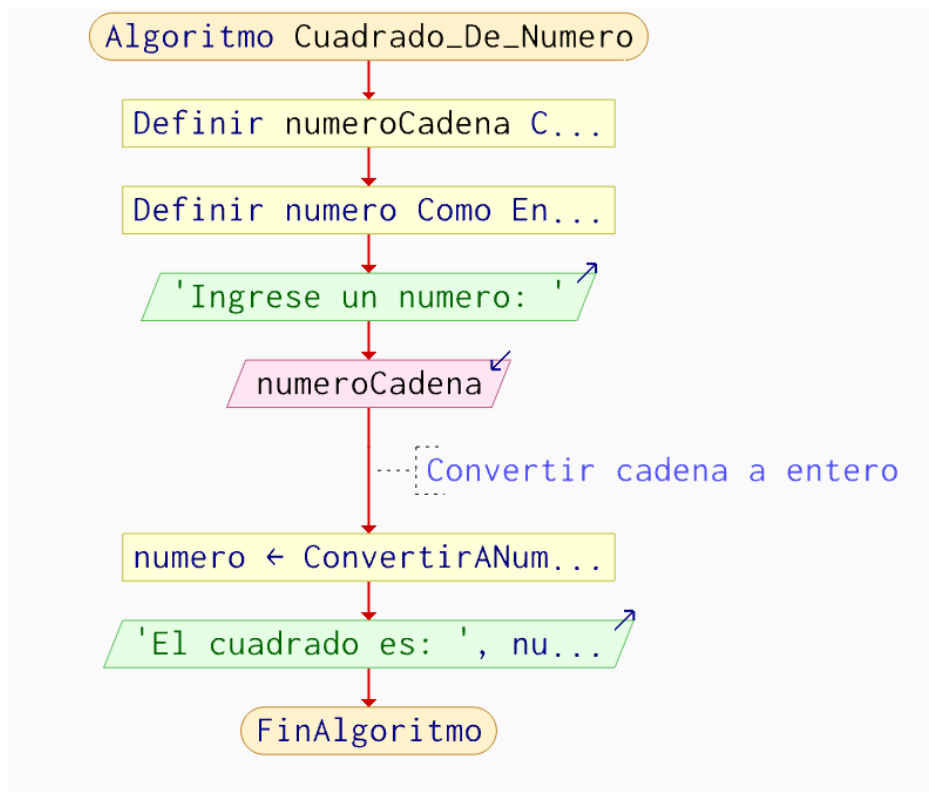
    printf("Ingrese un numero: ");
    fgets(numeroCadena, sizeof(numeroCadena), stdin);

    /* Convertir cadena a entero */
    numero = atoi(numeroCadena);

    printf("El cuadrado es: %d\n", numero * numero);

    return 0;
}
```

**Algoritmo** Cuadrado\_De\_Numero  
**Definir** numeroCadena **Como** Cadena  
**Definir** numero **Como** Entero  
**Escribir** "Ingrese un numero: "  
**Leer** numeroCadena  
  
**//** Convertir cadena a entero  
**numero** ← ConvertirANumero(numeroCadena)  
  
**Escribir** "El cuadrado es: ", numero \* numero  
**FinAlgoritmo**



[https://onlinegdb.com/xTckGH\\_y4H](https://onlinegdb.com/xTckGH_y4H)

## 6. Vamos invertir un vector y mostrar la parte original y la invertida

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

int main() {
    char texto[200];
    char invertida[200];
    int i, j = 0;

    printf("Ingrese una cadena: ");
    fgets(texto, sizeof(texto), stdin);
    texto[strlen(texto, "\n")] = '\0';

    /* Invertir la cadena */
    for (i = strlen(texto) - 1; i >= 0; i--) {
        invertida[j++] = texto[i];
    }
    invertida[j] = '\0';

    printf("Original: %s\n", texto);
    printf("Invertida: %s\n", invertida);

    return 0;
}
```

**Algoritmo** Invertir\_Cadena

**Definir** texto, invertida **Como** Cadena

**Definir** i, n **Como** Entero

**Escribir** "Ingrese una cadena: "

**Leer** texto

invertida ← ""

n ← Longitud(texto)

// Invertir la cadena

**Para** i ← n **Hasta** 1 **Con** Paso -1 **Hacer**

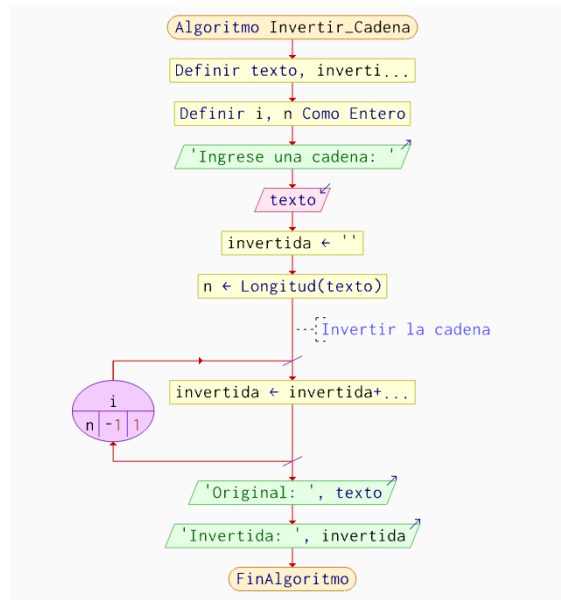
    invertida ← invertida + Subcadena(texto, i, i)

**FinPara**

**Escribir** "Original: ", texto

**Escribir** "Invertida: ", invertida

**FinAlgoritmo**



<https://onlinegdb.com/Q0uzH7l2s>

7. Vamos a crear una función que reciba una cadena analice y nos devuelva la cantidad de vocales

```
#include <stdio.h>
#include <string.h>
#include <ctype.h>

int contarVocales(char texto[]) {
    int contador = 0;
    int i;
    char c;

    for (i = 0; texto[i] != '\0'; i++) {
        c = tolower(texto[i]);

        if (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u') {
            contador++;
        }
    }

    return contador;
}

int main() {
    char texto[200];

    printf("Ingrese una cadena: ");
    fgets(texto, sizeof(texto), stdin);
    texto[strlen(texto)] = '\0';

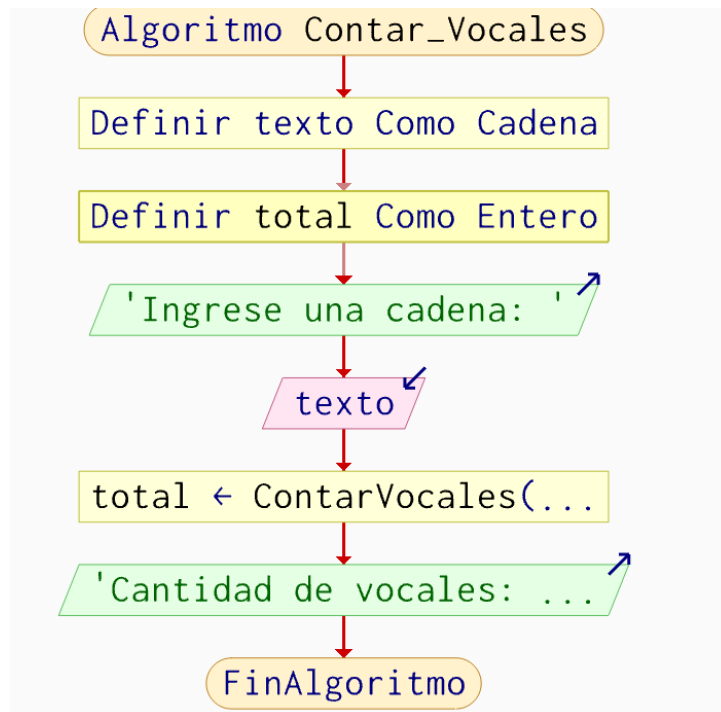
    printf("Cantidad de vocales: %d\n", contarVocales(texto));

    return 0;
}
```

```
Funcion
Inicio
    contador ← 0
    2! C = 'a' 0 C = 'e' 0 C = 'i' 0 C = 'o' 0 C = 'u' Entonces
        C ← tolower(texto[i])
        3! Si C == 'a' || C == 'e' || C == 'i' || C == 'o' || C == 'u' Entonces
            contador ← contador + 1
        FinSi
    FinRepetir
    FinFuncion

InicioPrograma
    Escribir "Cantidad de vocales: " total
    total ← contarVocales(texto)

    Leer texto
    Escribir "Ingrese una cadena: "
    Definir total Como Entero
    Definir texto Como Cadena
    Algoritmo contarVocales
```





<https://onlinegdb.com/wLbIm-QNV2>