

**Cuenta carácter**

**Brigada 6**

## **Análisis, describiendo el problema e identificando los datos de entrada y de salida**

Antes de empezar a determinar los datos de entrada y salida, así como el funcionamiento de nuestro programa, debemos de definir que es un arreglo ya que esto nos permitirá dar un gran paso significativo para la realización de nuestro proyecto. Por ello, podemos de finir al arreglo como un almacén, el cual nos permite tener diferentes cantidades o valores en una sola variable, las cuales se van a encontrar en posiciones de memoria continuas.

El objetivo de nuestro programa será contar las veces que se repite un carácter en una cadena y obtener un arreglo con la clave ASII de cada carácter, así como las veces n que se repite dicho carácter. De la misma manera, los parámetros que recibe la función se guardarán en el arreglo resultante.

A partir de esto, para que sea más productivo nuestro código, los datos de la palabra o cadena se obtendrán a partir del usuario, el cual deberá introducirlos a través del teclado para que se puedan leer en pantalla, para que después el programa traduzca este código en su lenguaje para que pueda realizar cada una de las acciones que se encuentran previstas en el código.

Después de que el usuario haya ingresado la cadena, el programa deberá evaluar mediante estructuras condicionales las veces en que se repite cada carácter, para ello se va a utilizar una función que nos permite contar el número de letras que tiene una cadena y a partir de esto hacer más fácil la evaluación de la condición. De la misma manera, a través de estructuras repetitivas se buscará traducir el carácter en código ASCII.

Finalmente, nuestro código deberá imprimir en pantalla el carácter en código ASCII y las veces que se repite el mismo.

De acuerdo con lo mencionado anteriormente, podemos definir que los datos de entrada estarán constituidos por la cadena que ingrese el usuario a través de su teclado, mientras que los datos de salida, serán el carácter traducido en código ASCII y las veces que se repite dicho carácter en la cadena.

## *Pseudocódigo*

INICIO

FUNC cuenta(palabra:CADENA, b[][2]:ENTERO) RET:vacio

x, contador:=0, z:=0, condic, cont1:=0, test:=0, a:=0:ENTERO

ESCRIBIR "Ingrese una cadena"

LEER palabra

condic:=longitud(palabra)

x:=0

MIENTRAS x<condic

MIENTRAS a<condic

SI x<>a ENTONCES

SI palabra[x]=palabra[a] ENTONCES

contador:=contador+1

cadena[a]:=0

FIN SI

FIN SI

a:=a+1

FIN MIENTRAS

SI palabra[x]<>0 ENTONCES

SI contador>=1 ENTONCES

b[z][1]:=contador+1

b[z][0]:=palabra[x]

test:=b[z][0]

FIN SI

DE LO CONTRARIO

b[z][1]:=1

b[z][0]:=palabra[x]

```
        test:=b[z][0]
    FIN DE LO CONTRARIO

    z:=z+1

    cont1:=cont1+1
FIN SI

contador:=0

x:=x+1
FIN MIENTRAS

x:=0
MIENTRAS x<cont1

    a:=0

    MIENTRAS a<2

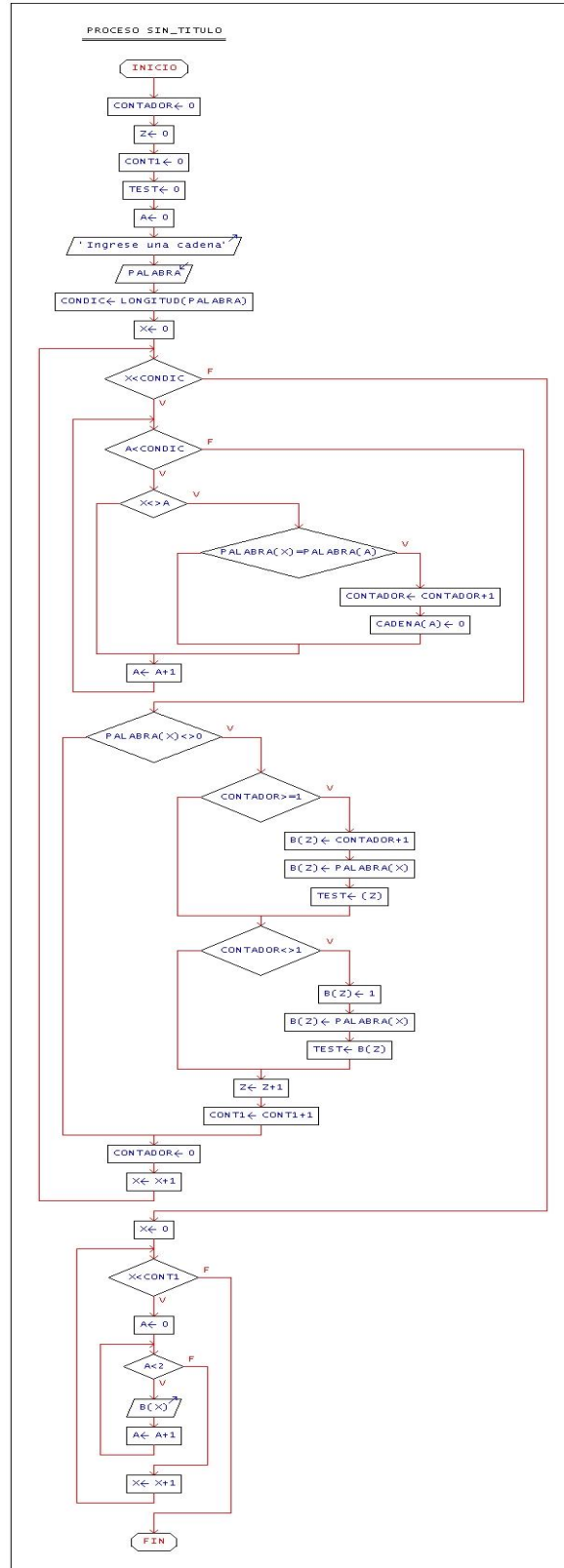
        ESCRIBIR b[x][a]

        a:=a+1

    FIN MIENTRAS

    x:=x+1
FIN MIENTRAS
FIN FUNC
FIN
```

## Diagrama de flujo



## *Código en C*

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

void contar(const char * palabra, int caracterArr[][2])
{
    int x, contador=0, z=0, condic, fac=0, test=0;
    int variableX=0;
    char* caracterB;
    printf("Ingrese una cadena \n");
    palabra = malloc (256);
    scanf("%255s", palabra );
    caracterB = strdup(palabra);
    condic=strlen(palabra);
    for (x=0; x<condic; x++)
    {
        while (variableX<condic)
        {
            if(x!=variableX && variableX<condic)
            {
                if (caracterB[x]==caracterB[variableX])
                {
                    contador=contador+1;
                    caracterB[variableX]=0;
                }
            }
            variableX++;
        }
        if (caracterB[x]!=0)
        {
            if (contador >= 1)
            {
                caracterArr[z][1]= contador + 1;
                caracterArr[z][0]= caracterB[x];
                test=caracterArr[z][0];
            }
            else
            {
                caracterArr[z][1]= 1;
                caracterArr[z][0]= caracterB[x];
                test=caracterArr[z][0];
            }
            z++;
            fac=fac+1;
        }
        contador=0;
    }
```

```

        variableX=0;
    }
    for (x=0 ; x<fac ; x++)
    {
        for (variableX=0 ; variableX<2 ; variableX++)
        {
            printf("%ld,",caracterArr[x][variableX]);
        }
        printf("\n");
    }
}

int main(){
    char * cadenita;
    int a2[1000][2];
    contar(cadenita, a2);
}

```

### **Pruebas de escritorio**

```

C:\Users\HP\Desktop\Hola_mundo\cuentaCaracteres\bin\Debug\cuentaCaracteres.exe
Ingrese una cadena
Facultad
70,1,
97,2,
99,1,
117,1,
108,1,
116,1,
100,1,

Process returned 7 (0x7)   execution time : 12.801 s
Press any key to continue.

Ingrese una cadena
Programación
80,1,
114,2,
111,1,
103,1,
97,2,
109,1,
99,1,
105,1,
-94,1,
110,1,

Process returned 10 (0xA)   execution time : 9.538 s
Press any key to continue.

```

```
main.c x
1  #include <stdio.h>
2  #include <string.h>
3
4  void contar(const char * palabra, int caracterArr[][2])
5  {
6      int x, contador=0, z=0, condic, fac=0, test=0;
7      int variableX=0;
8      char* caracterB;
9      printf("Ingrese una cadena \n");
10     palabra = malloc (256);
11     scanf("%255s", palabra );
12     caracterB = strdup(palabra);
13     condic=strlen(palabra);
14     for (x=0; x<condic; x++)
15     {
16         while (variableX<condic)
17         {
18             if(x!=variableX && variableX<condic)
19             {
20                 if (caracterB[x]==caracterB[variableX])
21                 {
22                     contador=contador+1;
23                     caracterB[variableX]=0;
24                 }
25             }
26             variableX++;
27         }
28         if (caracterB[x]!=0)
29         {
30             if (contador >= 1)
31             {
32                 caracterArr[z][1]= contador + 1;
33                 caracterArr[z][0]= caracterB[x];
34                 test=caracterArr[z][0];
35             }
36             else
37             {
38                 caracterArr[z][1]= 1;
39                 caracterArr[z][0]= caracterB[x];
40                 test=caracterArr[z][0];
41             }
42             z++;
43             fac=fac+1;
44         }
45         contador=0;
46         variableX=0;
47     }
```



```
48     for (x=0 ; x<fac ; x++)
49     {
50         for (variableX=0 ; variableX<2 ; variableX++)
51         {
52             printf("%ld,",caracterArr[x][variableX]);
53         }
54         printf("\n");
55     }
56 }
57
58 int main(){
59     char * cadenita;
60     int a2[1000][2];
61     contar(cadenita, a2);
62 }
63
```