

**CERO Y UNO**

**BRIGADA 6**

## **Análisis, describiendo el problema e identificando los datos de entrada y de salida “Cero y Uno”.**

### ➔ Problema:

Se nos solicita validar que una cadena que está formada por n ceros seguidos de n unos, debe regresar TRUE si la misma es válida pero si no debe regresar FALSE. Al igual que si se evalúa una cadena vacía debe de regresar FALSE.

### ➔ Análisis: Para esta sección debemos saber que datos vamos a introducir y que nos va a regresar:

-Datos de entrada: Se va a ingresar una cadena de tipo Carácter.

-Datos de salida: Nos va a regresar o mostrar un valor de tipo Lógico los cuales son

FALSE=0 TRUE=1

### ➔ Pseudocódigo: Al analizar el problema y lo que se nos solicita llegamos a que nuestro pseudocódigo necesitaría de estructuras repetitivas como For y de condición como If. Para que se analice cada elemento de la cadena de tipo carácter para que nos imprima el valor lógico de la misma.

## **Pseudocodigo**

**INICIO**

**y: ENTERO**

**x: ENTERO**

**Tam: ENTERO**

**Cad[]: CADENA**

**Tam:= strlen(Cad);**

**SI Tam %2 != 0 ENTONCES**

**ESCRIBIR "0"**

**FIN SI**

**SI Tam==0 ENTONCES**

**ESCRIBIR "0"**

**FIN SI**

**TamCero:= Tam/2: ENTERO**

**TamUno:= Tam/2: ENTERO**

**SI Cad[0] == "0" ENTONCES**

**y:=0**

**PARA (y:=0; y< TamCero-1; y=y+1)**

**SI Cad[y] != Cad [y+1] ENTONCES**

**ESCRIBIR "0"**

**FIN SI**

**FIN PARA**

**FIN SI**

**SI Cad[y+1]== "1" ENTONCES**

**PARA (y:=y+1; x<Tam-1; x=x+1)**

**SI Cad[x] != Cad[x+1] ENTONCES**

**ESCRIBIR "0"**

**FIN SI**

**FIN PARA**

**FIN SI**

**SI Cad [0]=="1" ENTONCES**

**y:= 0**

**PARA (y:=0; y< TamUno; y:=y+1)**

**SI Cad[y] != Cad[y+1] ENTONCES**

**ESCRIBIR "0"**

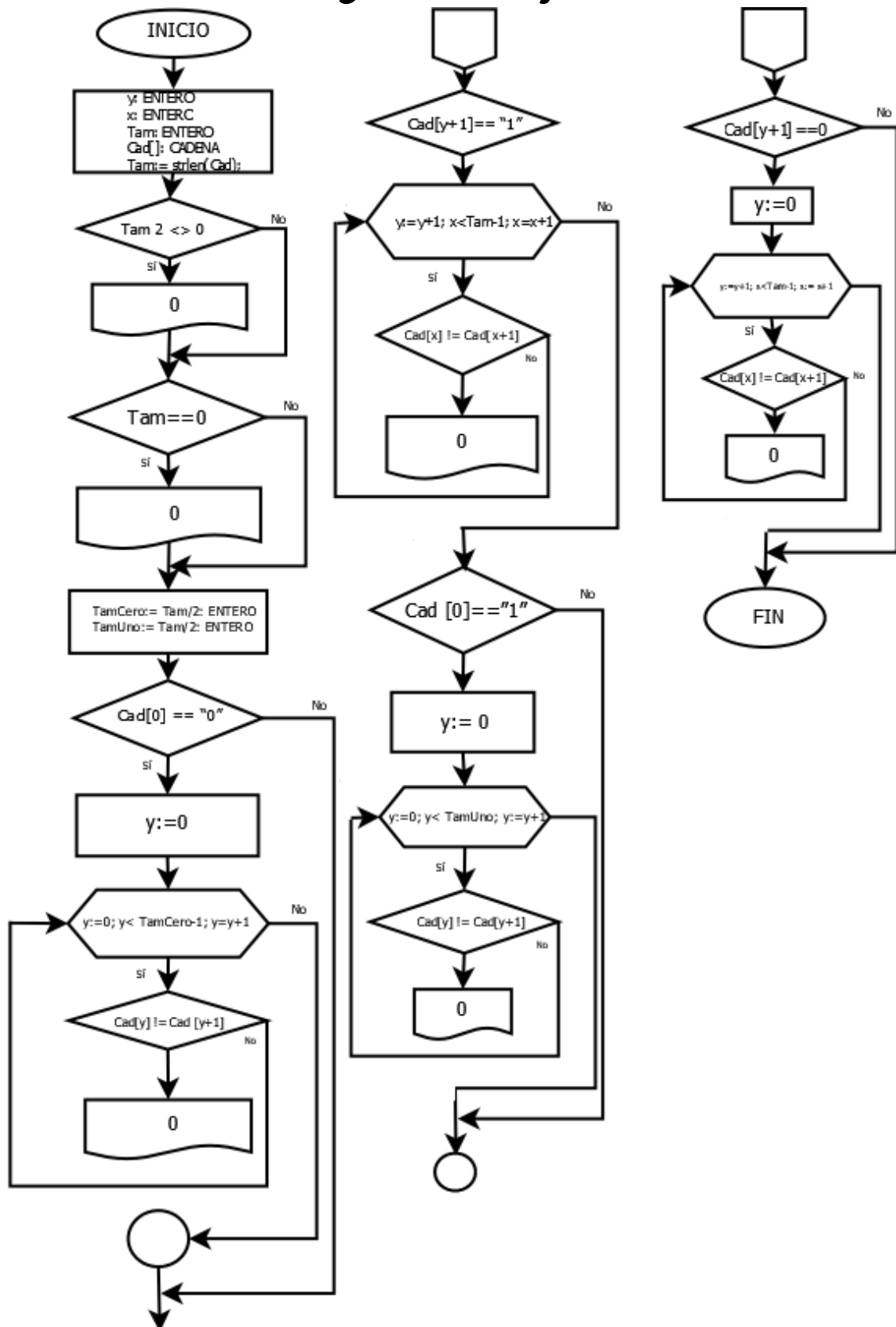
**FIN SI**

**FIN PARA**

**FIN SI**

```
SI Cad[y+1]==0 ENTONCES
  PARA (y:=y+1; x<Tam-1; x:= x+1)
    SI Cad[x] != Cad[x+1]
      ESCRIBIR "0"
    FIN SI
  FIN PARA
FIN SI
FIN
```

## Diagrama de flujo



## *Programa en C*

```
#include <stdio.h>
#include <string.h>
int validaCadena (const char *Cad)
{
    int Tam = strlen (Cad);
    if (Tam % 2 != 0)
    {
        printf ("%d\n", 0);
        return 0;
    }
    if (Tam == 0)
    {
        printf ("%d\n", 0);
        return 0;
    }
    int TamCero = Tam / 2;
    int TamUno = Tam / 2;
    if (Cad[0] == *"0")
    {
        int y = 0;
        for (y = 0; y < TamCero - 1; y++)
        {
            if (Cad[y] != Cad[y + 1])
            {
                printf ("%d\n", 0);
                return 0;
            }
        }
    }
}
```

```

    if (Cad[y + 1] == *"1")
    {
        for (int x = y + 1; x < Tam - 1; x++)
        {
            if (Cad[x] != Cad[x + 1])
            {
                printf ("%d\n", 0);
                return 0;
            }
        }
    }
}

else if (Cad[0] == *"1")
{
    int y = 0;
    for (y = 0; y < TamUno - 1; y++)
    {
        if (Cad[y] != Cad[y + 1])
        {
            printf ("%d\n", 0);
            return 0;
        }
    }

    if (Cad[y + 1] == *"0")
    {
        for (int x = y + 1; x < Tam - 1; x++)
        {
            if (Cad[x] != Cad[x + 1])
            {
                printf ("%d\n", 0);
            }
        }
    }
}

```

```

        return 0;
    }
}
}
}
printf ("%d\n", 1);
return 0;
}

```

## **PRUEBA DE ESCRITORIO JUNTO TEST**

```

1  if (Cad[y + 1] == *"1")
2  {
3      for (int x = y + 1; x < Tam - 1; x++)
4      {
5          if (Cad[x] != Cad[x + 1])
6          {
7              printf ("%d\n", 0);
8              return 0;
9          }
10     }
11 }
12 }
13 else if (Cad[0] == *"1")
14 {
15     int y = 0;
16     for (y = 0; y < TamUno - 1; y++)
17     {
18         if (Cad[y] != Cad[y + 1])
19         {
20             printf ("%d\n", 0);
21             return 0;
22         }
23     }
24     if (Cad[y + 1] == *"0")
25     {
26         for (int x = y + 1; x < Tam - 1; x++)
27         {
28             if (Cad[x] != Cad[x + 1])
29             {
30                 printf ("%d\n", 0);
31                 return 0;
32             }
33         }
34     }
35 }
36 printf ("%d\n", 1);
37 return 0;
38 }
39

```



```

#include <stdio.h>
#include <string.h>
#include <../ceroUno.c>

int main(){

valida Cadena ("");

validaCadena ("01");

valida Cadena ("0011");

validaCadena("000111");

validaCadena("011");

return 0;

}

```

```

0
1
1
1
0

```

## *TEST*

```

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

```

```

int main()
{
    Int validaCadena (const char * Cad);
    int cero=0;
    int uno=0;
    int y;
    int result;
    int Cad;

```

```
Cad=fopen("indexTestCeroUno.c","r");
```

```
for(y=0;y<30;y++)
```

```
{
```

```
    switch(Cad[y])
```

```
    {
```

```
        case '0':cero++;break;
```

```
        case '1':uno++;break;
```

```
    }
```

```
fclose(Cad);
```

```
}
```

```
if (cero!=uno || Cad(NULL))
```

```
    result=0;
```

```
else
```

```
    result=1;
```

```
}
```