

# CeroUno

## Objetivo del proyecto

Validar que una cadena está formada por n ceros seguidos de n unos.

Si es una cadena válida regresar TRUE. En caso contrario regresar FALSE.

Si se evalúa una cadena vacía o nula se debe regresar FALSE.

## Análisis del proyecto

El programa que se presenta valida las cadenas formadas por ceros sucedidos por unos, si la cadena no cumple estas condiciones regres FALSE, si la cadena es vacía regresa FALSE, TRUE es presentado en caso de que la cadena sea válida, se utilizan ciclos for.

## Pseudocódigo

INICIO

SUB PROCESO VCHAIN( \* CADCH)

SI (dim % 2 != 0)

ESCRIBIR ", 0

FIN SI

SI (dim==0)

ESCRIBIR ", 0

FIN SI

SI (cadch[0]==\*0)

PARA i=0 Hasta dimC-1 Con Paso 1

SI (cadch[i] != cadch[i+1])

ESCRIBIR ", 0

FIN SI

FIN PARA

SI (cadch[i+1]==\*1)

PARA int j=i+1 Hasta dim-1 Con Paso 1

SI (cadch[j] != cadch[j+1])

ESCRIBIR ", 0

FIN SI

FIN PARA

FIN SI

SI NO (cadch[0]==\*1)

PARA i=0 Hasta dimU-1 Con Paso 1

SI (cadch[i] != cadch[i+1])

ESCRIBIR ", 0

FIN SI

FIN PARA

SI (cadch[i+1]==\*0)

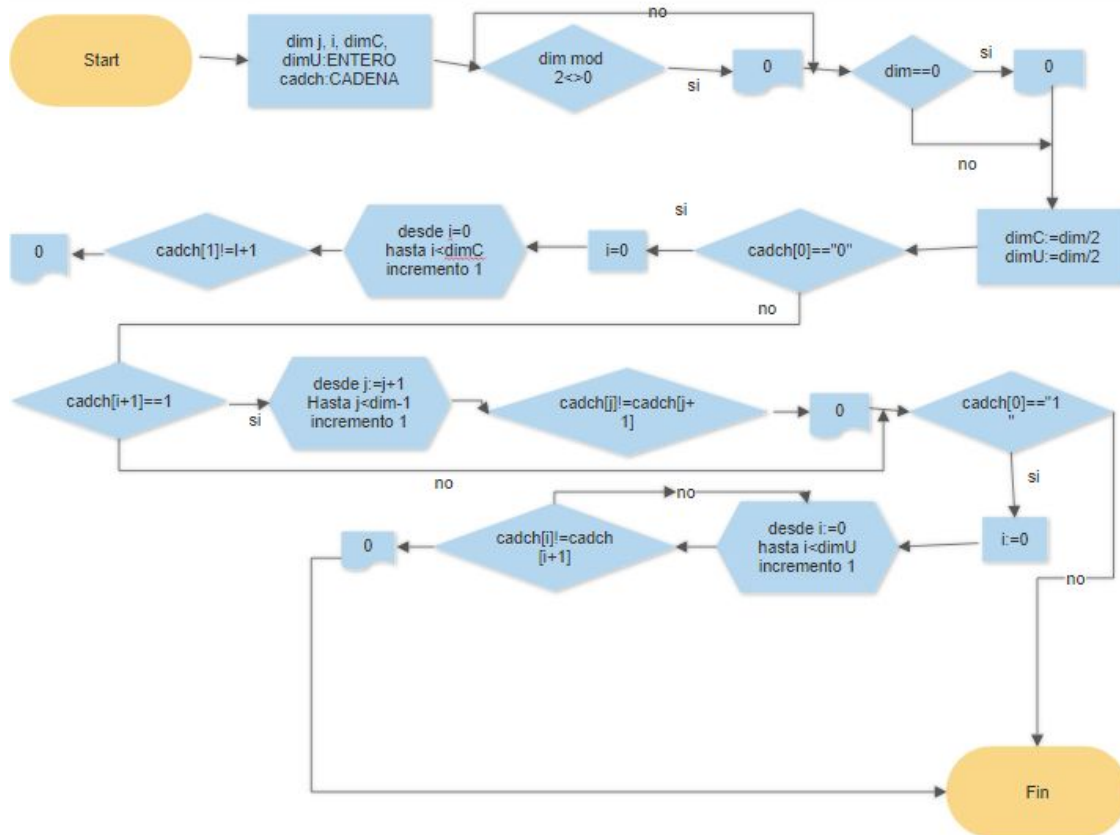
PARA int j=i+1 Hasta dim-1 Con Paso 1

SI (cadch[j] != cadch[j+1])

ESCRIBIR ", 0

FIN SI  
 FIN PARA  
 FIN SI  
 FIN SI  
 FIN SI  
 ESCRIBIR ", 1  
 FIN SUB PROECESO  
 FIN

## Diagrama de flujo



## codigo en c

```

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

int main()
{

int vchain(const char * cadch)
{
    if(cadch[0]==*"0"){
        int i=0;
        for(i=0; i< dimC-1; i++){

```

```

        if(cadch[i] != cadch[i+1]){
            printf("%d\n", 0);
            return 0;
        }

    }

    if(cadch[i+1]==*"1"){
        for(int j=i+1; j<dim-1; j++){
            if(cadch[j] != cadch[j+1]){
                printf("%d\n", 0);
                return 0;
            }
        }
    }
}

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

    if(cadch[i+1]==*"0"){
        for(int j=i+1; j<dim-1; j++){
            if(cadch[j] != cadch[j+1]){
                printf("%d\n", 0);
                return 0;
            }
        }
    }
}

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

int dim= strlen(cadch);

if(dim % 2 != 0){
    printf("%d\n", 0);
    return 0;
}

```

```
if(dim==0){
    printf("%d\n", 0);
    return 0;
}

int dimC = dim/2;
int dimU = dim/2;
}
printf("%d\n", 1);
return 0;

int dim= strlen(cadch);

if(dim % 2 != 0){
    printf("%d\n", 0);
    return 0;
}

if(dim==0){
    printf("%d\n", 0);
    return 0;
}

}
```

**test**

```
/tmp/NPsYfεqoTu.o
```

```
0011 TRUE|
```

**prueba de escritorio de código compilado**