**Alumna: Elizabeth Guerrero Torres Proyecto final Grupo: 1992**

**ÁNALISIS**

* Dada una cadena, obtener su código Ascii y mostrar su equivalente en binario.

**OBJETIVO**

* Mostrar el código binario de una cadena, la cual es proporcionada por el usuario.

**ENTRADA**

* sLetter[50]: variable de tipo string que almacenará el mensaje de la cadena dada por el usuario. El 50 significa que puede guardar hasta esa cantidad de caracteres.

**PRE-CONDICIONES**

* eCount será una variable contador de tipo entero que irá aumentando de uno en uno.
* eReverse será una variable de tipo entero que irá disminuyendo su hasta que queden ordenados los números según el código binario.
* eResult[8] será un arreglo de tipo entero que almacenará el módulo del número ascii dividido entre 2.
* A pesar de que la variable de entrada sLetter está declarada como un string, sólo recibe un carácter.

**RESTRICCIONES**

* Sólo se aceptará un carácter.

**SALIDA**

* Imprimir eResult[eReverse]

**PSEUDOCÓDIGO**

1. Inicio
2. Inicializar variables:
   1. sLetter[50] 🡨 "o"
   2. eCount 🡨 0
   3. eReverse 🡨 0
   4. eResult[8] 🡨 {0}
3. Imprimir “Escribe una letra: ”
4. Leer sLetter
5. Mientras(eCount < 8)
   1. eResult[eCount] 🡨 sLetter[0] % 2
   2. sLetter[eCount] 🡨 sLetter[eCount] / 2
   3. eCount++
6. Fin Mientras
7. Para(eReverse = eCount – 1; eReverse >= 0; eReverse--)
   1. Imprimir eResult[eReverse]
8. Fin Para
9. Fin.

**DIAGRAMA DE FLUJO**

**Entrada:**

* **sLetter[50]: string**

**Inicializar variables:**

* **sLetter[50] 🡨 ‘o’**
* **eCount 🡨 0**
* **eReverse 🡨 0**
* **eResult[8] 🡨 {0}**

**Imprimir “Escribe una letra: ”**

**1**

**1**

**Leer sLetter**

**¿eCount < 8?**

**Mien**

**SÍ**

**eResult[eCount] 🡨 sLetter[0] % 2**

**NO**

**sLetter[eCount] 🡨 sLetter[eCount] / 2**

**eCount++**

**1**

**1**

**eReverse 🡨 eCount - 1**

**¿eReverse >= 0?**

**SÍ**

**Imprimir eResult[eReverse]**

**NO**

**eReverse --**

|  |  |
| --- | --- |
| Diagrama N-S | |
| Inicio | |
| Entrada:   * sLetter[50] 🡨 string; | |
| Inicializar variables:   * sLetter[50] 🡨 “o” * eCount 🡨 0 * eReverse 🡨 0 * eResult[8] 🡨 {0} | |
| Imprimir “Escribe una letra: ” | |
| Leer sLetter | |
|  | **eResult[eCount] 🡨 sLetter[0] % 2** |
| **sLetter[eCount] 🡨 sLetter[eCount] / 2** |
| **eCount++** |
| Mientras(eCount < 8) | |
| Para eReverse 🡨 eCount – 1 hasta eReverse >= 0 eReverse--  **Imprimir eResult[eReverse]** | |
| Fin | |

|  |  |  |
| --- | --- | --- |
| Prueba de escritorio | | |
| Prueba | **Datos de entrada** | **Salida** |
| **sLetter** | **eResult[eReverse]** |
| 1 | **h** | **01101000** |
| 2 | **a** | **01100001** |
| 3 | **l** | **01101100** |
| 4 | **o** | **01101111** |