

U.T. 2.- Introducción a la Programación estructurada y modular en los lenguajes de script clientes y servidor

Nombre del Alumno: _____

Calificación Final de la PRUEBA: _____

MB=1

Cuestión 1.-

Corrige los errores no sintácticos que provoca que el código que se proporciona a continuación no realice la acción indicada.

```
//Dado una secuencia de caracteres determinar cuantas letras minúsculas tiene.
var
    parrafo: cadena
    indice: numerico
    contador: numerico
inicio
    imprimir("\n Introduce una secuencia de caracteres..... ")
    leer(parrafo)
    indice=1
    mientras(indice<strlen(parrafo))
    {
        si(parrafo[indice]>'a' and parrafo[indice]<'z')
        {
            contador=contador+1
            indice=indice+1
        }
    }
    imprimir("\n El parrafo [",parrafo,"] tiene ", contador," minúsculas")
fin
```

□
MB=1

Cuestión 2.-

Rellena el espacio para completar el código que se proporciona.

```
/* Ejercicio 49 Dado un vector de secuencias determinar las que son palíndromo. */  
  
var  
    palabra : vector [*] cadena  
    longitud : numerico  
    indice : numerico  
    indicev : numerico  
    tama: numerico  
    palabra_aux: cadena  
  
// FIN DE DECLARACION DE LAS VARIABLES  
  
inicio  
cls() // SIMPLE LIMPIADOR DE LA VENTANA EJECUTABLE  
  
    imprimir ("\nCuantas palabras vas a introducir?")  
    leer (tama)  
    dim(palabra,tama)  
  
    indicev=1  
    mientras (indicev<=tama)  
    {  
        imprimir ("\nDime la frase ", indicev)  
        leer (palabra[indicev])  
        indicev=indicev+1  
    }  
  
    indicev=1  
    mientras (indicev<=tama)  
    {  
        longitud=strlen(palabra[indicev])  
        palabra_aux=palabra[indicev]  
        indice=1  
        mientras (                                     )  
        {  
  
        }  
        si (                                     )  
        {  
            imprimir ("\nLa palabra ", palabra_aux, " es un palindromo")  
        sino  
            imprimir ("\nLa palabra ", palabra_aux, " no es un palindromo")  
        }  
        indicev=indicev+1  
    }  
  
fin
```

□
MB=1

Cuestión 3.-

Determina el enunciado del código que se presenta.

/*

*/

```
// La función ord() tiene como parámetro un carácter y return el código ascii (entero/posición en la tabla ascii)  
// La función ascii() recibe como parámetro un valor numérico enteral natural y devuelve el carácter asociado considerando la tabla ascii
```

```
var
```

```
    a: cadena
```

```
    b: cadena
```

```
    c: numerico
```

```
    d: numerico
```

```
    e: numerico
```

```
    f: numerico
```

```
inicio
```

```
    imprimir("\n Introduce una secuencia de caracteres..... ")
```

```
    leer(a)
```

```
    b=a
```

```
    d=strlen(a)
```

```
    f=1
```

```
    mientras(f<=int(d/2))
```

```
    {
```

```
        e=ord(a[f])
```

```
        a[f]=a[d-f+1]
```

```
        a[d-f+1]=ascii(e)
```

```
        f=f+1
```

```
    }
```

```
    imprimir("\n La inversa del a ["a"] es ", b)
```

```
fin
```

□	□	□	□	□		Cuestión 4.-
M=0	$R_{\downarrow}=0,25$	R=0,75	B=1	MB=1,5		

Números amigos: Son parejas de números tales que la suma de los divisores de uno da el otro, y viceversa. Los más conocidos son el 284 y el 220, ya que los divisores de 284: $1+2+4+71+142=220$, y los divisores de 220: $1+2+4+5+10+11+20+22+44+55+110=284$

Números sociables: Lo mismo de antes, pero en vez de ser en parejas, es en grupos más grandes. Se suelen poner en secuencia, y la suma de los divisores del primero dan el segundo, los del segundo el tercero, y así hasta volver al primero. Por ejemplo, son números sociables el conjunto formado por 1264460, 1547860, 1727636, 1305184.

Problema: Dado una secuencia de n números (almacenados en un vector) determinar si son sociables

