

Práctico 7: Ejercicios de Repaso (tipo parcial)

1) Dado el arreglo MuchasLetras de MaxLetras caracteres que está totalmente completo con caracteres alfabéticos sin blancos y un número pedido por teclado en CantLetras, eliminar todas las letras que se encuentran en MuchasLetras exactamente CantLetras veces y dejar todos los lugares que ocupaban al principio del arreglo con blancos sin modificar el orden de las restantes letras.

2) Se tiene un arreglo ArrEnteros de enteros positivos de longitud MaxEnt que forman secuencias, separadas por uno o más ceros. Se desea buscar la secuencia menor **en cuanto a la sumatoria de sus elementos** y trasladarla al final del arreglo, junto con TODOS los ceros que hay a su izquierda, haciendo un corrimiento a izquierda. Si se detecta más de una secuencia menor, considerar la primera. Nota: El arreglo empieza siempre con uno ó más ceros y YA tiene los datos cargados. Realice el DE y codifique la solución. No se puede utilizar estructuras auxiliares.

EJEMPLO, si se recibe este arreglo de entrada:

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 9 | 9 | 0 | 0 | 9 | 3 | 1 | 2 | 0 | 0 | 7 | 7 | 0 | 0 | 0 | 0 | 9 | 8 | 1 | 1 | 3 | 0 | 0 | 0 | 3 | 6 | 7 | 9 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

El arreglo quedaría:

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 9 | 9 | 0 | 0 | 9 | 3 | 1 | 2 | 0 | 0 | 0 | 0 | 9 | 8 | 1 | 1 | 3 | 0 | 0 | 0 | 3 | 6 | 7 | 9 | 0 | 0 | 0 | 7 | 7 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

3) Dados dos arreglos de caracteres, TEXTO y PATRÓN, realizar un programa que detecte la presencia de PATRÓN dentro de TEXTO . Si lo encuentra lo borra realizando un corrimiento a izquierda y llenando los espacios finales en blanco.

4) **(Parcial 2017)** Se tiene el arreglo *Números* con *Max* enteros en el que ya están cargadas secuencias de números enteros positivos separados por uno o más ceros entre ellas. El arreglo puede o no comenzar y terminar con uno o más ceros.

a) Implemente un módulo *DevuelveSecuencia* que recibe como parámetros a Números, una pos i, y devuelve la posición inicial y final de la próxima secuencia a partir de i.

b) Implemente un módulo *Sumatoria*, que recibe a Números, posi, posf y retorna la sumatoria de los enteros que están entre ambas posiciones

c) Usando los módulos anteriores y otros nuevos, implemente la solución completa para el siguiente problema: Dado un número entero positivo *SumaBuscada* solicitado al usuario detectar todas las secuencias cuya suma sea igual a ese número e invertir sus elementos manteniendo la secuencia su posición en el arreglo.

Realice programa principal y DE. No usar estructuras auxiliares. Asuma que el arreglo ya está cargado.

5) **(Parcial 2019)** Debe hacer el Diagrama de Estructura y codificar en Pascal el programa principal (con constantes, tipos y variables) e implementar el procedimiento que:

- Dado un arreglo completo con caracteres llamado LETRAS de longitud MaxLET,
- y otro también completo de caracteres llamado MODELO de MaxMOD (con MaxMOD < MaxLET)

Desplace hasta el final del arreglo sólo la primer ocurrencia de MODELO en el arreglo LETRAS (si es que está) corriendo los caracteres siguientes hacia adelante. (Asuma en el programa principal que existen los

procedimientos de carga de arreglos y para mostrar los resultados, es decir, no los implemente). Puede modularizar el procedimiento todo lo que desee. No puede utilizar arreglos auxiliares.

| | | |
|---|--|--|
| <u>Ejemplo 1:</u> LETRAS: {DCDEFJKLL} Modelo: DEF Salida: (está 1 vez) LETRAS {DCJKLLDEF} | <u>Ejemplo 2:</u> LETRAS: {DCDEFJKLL} Modelo: DEA Salida: (no está) LETRAS {DCDEFJKLL} | <u>Ejemplo 3:</u> LETRAS: {DCDEFRDEM} Modelo: DE Salida: (está 2 veces) LETRAS {DCFRDEMDE} |
|---|--|--|

6) **(Parcial 2018)** Debe escribir el programa principal (con constantes tipos y variables) e implementar el procedimiento que:

- Dado un arreglo de caracteres llamado LETRAS de longitud MaxLetras que posee palabras pegadas. (Tiene blancos al final hasta completar el arreglo),
- Dado un arreglo de caracteres llamado PALABRA de MaxLetrasPalabra (con MaxLetrasPalabra < MaxLetras) que posee una palabra (Tiene blancos al final hasta completar el arreglo),
- Dada también una variables con un valor entero llamada POS.

Si PALABRA NO está en LETRAS, agregarlo a partir de la posición POS y desplazando a derecha las letras siguientes de LETRAS para hacer el espacio necesario. Se asegura que hay lugares libres en LETRAS.

(Asuma en el programa principal que existen los procedimientos de carga de arreglos y para mostrar los resultados, es decir, no los implemente)

| | |
|---|---|
| <u>Ej 1:</u> (MaxLetras=20,MaxLetrasPalabra=4) [CAMPOSOLPERROVACA] PALABRA: [SOL] POS: 15 [CAMPOSOLPERROVACA] <i>(Estaba, no se hizo nada)</i> | <u>Ej 2:</u> (MaxLetras=24,MaxLetrasPalabra=4) [CAMPOSOLPERROVACA] PALABRA: [CASA] POS: 4 [CAMCASAPOSOLPERROVACA] <i>(No estaba, se agregó CASA en la posición 4)</i> |
|---|---|

7) **Recuperatorio (2018)** Se tiene el arreglo *numeros* de **MAXNUM** enteros ya cargado (el arreglo está completamente cargado con números positivos entre 1 y 9 comenzando en la primer posición hasta la posición **MAXNUM**). Se pide que realice el Diagrama de Estructura y el programa principal, declaraciones de constantes, tipos, variables y módulos en Pascal y para resolver lo siguiente:

Detectar todos los números consecutivos cuya suma sea igual a **valor** y reemplazar la primera de esas celdas por **valor** y las restantes por cero. Luego todos los ceros deben quedar al principio del arreglo y el resto de los números deben mantener su orden original. **valor** es un número entero que se le debe solicitar al usuario y debe controlar que el mismo sea mayor o igual a 10, de lo contrario solicitarlo de nuevo hasta que ingrese un valor válido.

Ej: **valor** = 10 **Arreglo original:** 3 5 6 7 3 4 5 1 8 **Arreglo resultante final:** 0 0 0 3 5 6 10 10 8
(MAXNUM=9)

Ej: **valor** = 11 **Arreglo original:**1 2 3 5 6 2 7 9 9 **Arreglo resultante final:** 0 0 0 11 6 2 7 9 9
(MAXNUM=9)

8)Se tiene un array de chars de longitud max llamado **ArrChar**

- 1) Implementar un módulo (función o procedimiento según corresponda) que reciba el arreglo **ArrChar**, una posición **pos** (integer entre 1 y max) y una cantidad **cant**. El módulo debe eliminar la cantidad indicada por **cant** de caracteres desde la posición **pos** inclusive haciendo un corrimiento a izquierda. Al final debe completar esa cantidad **cant** con blancos. (**cant** indica un valor válido dentro de los límites del arreglo a partir de la posición **pos**)

- 2) Haciendo uso del módulo anterior y todos los que considere necesario, hacer un programa que dado el arreglo **ArrChar**, busca todas las apariciones consecutivas de una misma letra, eliminando todas las repeticiones con un corrimiento hacia izquierda y dejando sólo la primera aparición de la letra.

Debe realizar el Diagrama de Estructuras, los módulos pedidos y otros que requiera así como el programa principal con las declaraciones de constantes, tipos y variables.

Ejemplo: ArrChar

| | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| c | c | c | a | j | u | k | b | l | l | l | l | a | n | a | r | e | z | u | l | a | r |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

Salida de ArrChar

| | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|--|--|--|--|--|
| c | a | j | u | k | b | l | a | n | a | r | e | z | u | l | a | r | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|--|--|--|--|--|

- 9) Dos amigos necesitan enviarse por mensajes muy importantes y extremadamente reservados, por lo que deciden decodificar los mensajes. La estrategia que utilizan es la siguiente: Ciertas partes del texto disjuntas (o sea sin superposición) las escriben en orden inverso y los encierran entre paréntesis, de manera tal de no olvidar que esos trozos deben ser leídos al revés. Para facilitarles la tarea se te pide que definas el DE y escribas un programa que permita decodificar los mensajes enviados. El mensaje a decodificar se encuentra en un array de chars **SECRETOIN** y se guardará el mensaje decodificado en otro array **SECRETOOUT**. **Aclaraciones:** El mensaje original a transmitir está formado por caracteres que pueden ser letras, números y/o signos de puntuación pero no paréntesis. De esta manera se evitan problemas en el momento del encriptado. Los paréntesis están bien colocados. Los paréntesis de apertura y de cierre serán reemplazados por un espacio en blanco.

Ejemplo Si SECRETOIN contiene:

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|--|---|---|---|--|---|---|---|---|---|---|---|---|---|--|---|---|---|---|---|--|---|---|---|---|
| H | O | Y | (| . | S | H | 2 | 2 | | S | A | L | | A |) | (| O | R | E | P | S | E | | E | T |) | E | N | | C | A | S | A |
|---|---|---|---|---|---|---|---|---|--|---|---|---|--|---|---|---|---|---|---|---|---|---|--|---|---|---|---|---|--|---|---|---|---|

SECRETOOUT deberá contener:

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|--|---|--|---|---|---|--|---|---|---|---|---|--|--|---|---|--|---|---|---|---|---|---|--|---|---|--|---|---|---|--|
| H | O | Y | | A | | L | A | S | | 2 | 2 | H | S | . | | | T | E | | E | S | P | E | R | O | | E | N | | C | A | S | |
|---|---|---|--|---|--|---|---|---|--|---|---|---|---|---|--|--|---|---|--|---|---|---|---|---|---|--|---|---|--|---|---|---|--|

- 10) (**Recuperatorio 2014**) Se tiene el arreglo FRASE de MAXLETRAS caracteres con un conjunto de palabras separadas por un espacio en blanco, debe justificarlas (redisponer las palabras de tal forma que queden distribuidas con igual cantidad de blancos entre ellas y ocupando todo el arreglo).

Ejemplo, si FRASE tiene:

| | | | | | | | | | | | | | | | | | | | |
|---|---|--|---|---|---|--|---|---|---|---|--|--|--|--|--|--|--|--|--|
| E | S | | U | N | A | | C | A | S | A | | | | | | | | | |
|---|---|--|---|---|---|--|---|---|---|---|--|--|--|--|--|--|--|--|--|

Se distribuyen quedando 5 espacios antes y 6 después ya que la cantidad es impar.

| | | | | | | | | | | | | | | | | | | | |
|---|---|--|--|--|--|--|---|---|---|--|--|--|--|--|--|---|---|---|---|
| E | S | | | | | | U | N | A | | | | | | | C | A | S | A |
|---|---|--|--|--|--|--|---|---|---|--|--|--|--|--|--|---|---|---|---|

Notas: Se asegura que la primer palabra comienza en la primer posición y la cantidad de blancos es adecuada para una distribución uniforme (igual cantidad de espacios entre todas las palabras). No puede utilizar estructuras auxiliares.