

Práctico 8: Ejercicios tipo Parcial

Resolver los siguiente problemas, realizando el DE y codificando la solución, SIN utilizar estructuras auxiliares y asumiendo que los datos de entrada ya están cargados

1) Se tiene un arreglo de caracteres llamado LETRAS de longitud MaxLET, y otro arreglo también de caracteres llamado MODELO de MaxMOD (con $\text{MaxMOD} < \text{MaxLET}$). Ambos arreglos están completos con caracteres distintos del espacio en blanco. Se pide: Desplazar hasta el final sólo la primer ocurrencia de MODELO en el arreglo LETRAS (si es que está) corriendo los caracteres siguientes hacia adelante. Realizar todo el código Pascal con la definición del programa principal, constantes, tipos y variables y todos los módulos que considere necesarios. Asuma en el programa principal que existen los procedimientos de carga de arreglos y para mostrar los resultados, (es decir, no los codifique). No puede utilizar arreglos auxiliares.

Ejemplo 1: LETRAS: {DX DAF JKLL} Modelo: DAF Salida: (está 1 vez) LETRAS {DXJKLL DAF }	Ejemplo 2: LETRAS: {DCDEFJKLL} Modelo: DEC Salida: (no está) LETRAS {DCDEFJKLL}	Ejemplo 3: LETRAS: {DCDEFRDEM} Modelo: DE Salida: (está 2 veces, toma la primera) LETRAS {DCFRDEM DE }
---	--	--

2) 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. Se pide resolver, dado un número entero positivo *SumaBuscada* solicitado al usuario detectar todas las secuencias cuya suma sea igual a ese número y ordenar sus elementos de menor a mayor manteniendo la secuencia su posición en el arreglo. Se debe utilizar el método de ordenamiento por inserción

3) Dos amigas necesitan enviarse mensajes extremadamente reservados. Ante el temor de que alguien pueda leer los correos, deciden codificarlos. 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. El mensaje a decodificar se encuentra en un array de chars **MENSAJEIN** 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	A
---	---	---	--	---	--	---	---	---	--	---	---	---	---	---	--	--	---	---	--	---	---	---	---	---	---	--	---	---	--	---	---	---	---

4) Se tiene un arreglo **ArrPalabras** de chars de tamaño MAXCAR cuyos caracteres forman palabras. Las palabras en el arreglo no están separadas por ningún carácter especial ni están ordenadas por ningún criterio. Los lugares vacíos del arreglo están marcados con un carácter de espacio en blanco y todos juntos al final del mismo. Para determinar dónde termina una palabra y comienza la siguiente, se cuenta con otro Arreglo **LongPalabras** donde cada elemento indica la longitud de la palabra que se encuentra en esa posición en el arreglo. En el Ejemplo se ve que la primera palabra tiene 4 posiciones (casa), la segunda palabra 6 posiciones (mueble) y así sucesivamente.

ArrPalabras

c	a	s	a	m	u	e	b	l	e	p	l	a	z	a	c	a	m	p	o
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

LongPalabras: 4 6 5 5

Resolver: dados una palabra (que es un arreglo **PalabraNueva** de chars con longitud máxima MAXPAL y posiciones en blanco al final) y una posición(integer) inserte esa palabra en el arreglo **ArrPalabras** en la posición indicada, actualizando el arreglo **LongPalabras**. No deben quedar blancos. Por ejemplo: si la posición dada es 2, la nueva palabra se debe insertar luego de la palabra “casa” y antes de “mueble”.

Ejemplo: Dado **PalabraNueva** o s o y posición:2

Resultado: **ArrPalabras**

c	a	s	a	o	s	o	m	u	e	b	l	e	p	l	a	z	a	c	a	m	p	o
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

LongPalabras : 4 3 6 5 5

5) 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). Detectar la última serie de números consecutivos cuya suma sea igual a VALOR. Si se la detecta se la debe duplicar pegada a la derecha de la serie encontrada, haciendo un desplazamiento a la derecha de los números siguientes conservando su orden y perdiendo los últimos. Si no hay lugar para realizar la duplicación no se hace la duplicación y avisa por pantalla. 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 6 4 5 1 8 4 1 8 9

Arreglo resultante final: 3 5 6 7 3 6 4 5 1 4 5 1 8 4

Ej: valor = 10 Arreglo original: 8 7 8 3 5 6 7 3 6 4 5 1 8 9

Arreglo resultante final: 8 7 8 3 5 6 7 3 6 4 5 1 8 9

- “Faltó Espacio para duplicar”-

6) Dados un arreglo de caracteres, TEXTO, de dimensión N (totalmente cargado) y otro arreglo de caracteres PATRÓN de dimensión M (con $M \leq N$), realizar un procedimiento que si encuentra PATRÓN dentro de TEXTO, lo ordena alfabéticamente sobre si mismo. Se debe utilizar el método de ordenamiento por selección

7) 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.

ejemplo

Para este arreglo de entrada: a b c d e f g h i j a c e a f f x r w x m x o z

y CantLetras=3

la salida sería: - - - - - b c d e f g h i j c e x r w x m x o z

8) Dado un arreglo FRASE de caracteres (de longitud MaxFrase) que está compuesto por palabras separadas por uno o más espacios en blanco. Realizar el Diagrama de Estructuras y el código que invierta todas las palabras que NO son palíndromas y reemplace por el símbolo “*” las letras de las palabras palíndromas. (Una palabra palíndroma es aquella que se lee igual de izquierda a derecha que de derecha a izquierda. Ej; “oso”, “asa”, cualquier letra sola, etc.)

Notas: El arreglo puede empezar y terminar con uno o más blancos.

Realice el código completo. Asuma que están implementados los procedimientos cargar(frase) e imprimir(frase), es decir, invoquelos pero no los implemente. Implemente todas las funciones y procedimientos que considere necesarios. No puede utilizar estructuras auxiliares.

Ejemplo:

Si se recibe esta frase: “y este es un caso distinto que ese de alla”

Deberá quedar así: “* etse se nu osac otnitsid euq *** ed *****”

9) Dado un arreglo ARR_SEC de caracteres con secuencias (donde una secuencia de caracteres es un serie consecutiva de chars diferentes a blancos) ordenar descendentemente el arreglo en base a la longitud de sus secuencias de ARR_SEC usando el método de ordenamiento de burbujeo. Nota el arreglo puede ó no empezar y terminar con blancos. Entre secuencias del arreglo puede haber uno ó mas blancos.