

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 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 .

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
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

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.

- 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*.
- Implemente un módulo *Sumatoria*, que recibe a *Números*, *posi*, *posf* y retorna la sumatoria de los enteros que están entre ambas posiciones
- 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 y ordenar sus elementos de menor a mayor manteniendo la secuencia su posición en el arreglo.

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) Se tiene un arreglo SECNUMEROS 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 colocarla al principio del arreglo. Si se detecta más de una secuencia menor, considerar la primera.

Nota: El arreglo puede ó no empezar y terminar con ceros.

EJEMPLO,

Si se recibe este arreglo: 0 9 9 0 0 9 3 1 9 0 0 0 9 7 9 0 0 0 0 **3 5 2 1 4** 0 0 0 9 6 7 9 1

El arreglo quedaría: **3 5 2 1 4** 0 9 9 0 0 9 3 1 9 0 0 0 9 7 9 0 0 0 0 0 0 0 9 6 7 9 1