

Examen - Agosto de 2002 - 2ª parte

- No se podrá utilizar ningún tipo de material (apuntes, libro, calculadora, etc).
- **Sólo** se contestarán preguntas sobre interpretación de la letra hasta 30 minutos antes de la finalización del mismo.
- Escriba las hojas de un sólo lado
- Las partes no legibles del examen se considerarán no escritas
- En la primer hoja a entregar ponga con letra clara, en el ángulo superior derecho, salón en el cual desarrolló la prueba, su nombre, número de cédula de identidad y cantidad de hojas -en ese orden-; las demás hojas es suficiente con nombre, número de cédula y número de página.
- Al entregar su prueba recuerde firmar la planilla correspondiente

Problema 1	30 pts	
-------------------	---------------	--

Se desea implementar una variante del juego *Sopa de Letras* :

Dada una palabra y una matriz de caracteres (la sopa), deben encontrarse todas las ocurrencias de esa palabra en la matriz, leyendo en dirección horizontal, vertical, y ambas diagonales, y considerando ambos sentidos (hacia arriba y hacia abajo; de abajo a la izquierda hacia arriba a la derecha y viceversa; etc.) en todos los casos.

Se pide :

a) Implemente las funciones:

- `function v = obtenerVectorArriba (M, i, j)`
- `function v = obtenerVectorIzquierda (M, i, j)`
- `function v = obtenerVectorAbajoALaDerecha (M, i, j)`
- `function v = obtenerVectorArribaALaDerecha (M, i, j)`

que dadas la matriz de caracteres M, y dos coordenadas (i, j), retornan el vector v que se construye recorriendo M en la dirección especificada por la función, sea o no una palabra con sentido.

Esta recorrida finaliza cuando no quedan más elementos de la matriz a recorrer en esa dirección y sentido.

Ejemplo :

Si M es

	1	2	3	4	5	6
1	X	S	H	A	A	T
2	G	A	J	I	N	B
3	U	C	L	E	R	O
4	L	K	H	O	Z	L
5	S	A	L	M	H	P
6	E	T	W	O	T	E
7	I	R	U	P	A	V

entonces :

- `obtenerVectorDerecha (M, 1, 3)` → ['H', 'A', 'A', 'T']
- `obtenerVectorIzquierda(M, 5, 6)` → ['P', 'H', 'M', 'L', 'A']
- `obtenerVectorAbajoALaDerecha (M, 7, 3)` → ['U']
- `obtenerVectorAbajo (M, 6, 2)` → ['T', 'R']

Aclaraciones:

- Asuma que las coordenadas dadas son correctas respecto al tamaño de la matriz.
- Son válidos los vectores de un (1) elemento

b) Implemente la función

- `function res = prefijo (w, v)`

que dados los vectores w y v, retorna:

- `res = 1`, si w es prefijo de v, esto es, si w es de largo `wlength`, entonces w coincide con los primeros `wlength` caracteres de v.
- `res = 0` en caso contrario.

Aclaración:

- No haga comparaciones innecesarias si, conociendo el tamaño de los vectores, se deduce que w no puede ser prefijo de v.
- No puede utilizar funciones de comparación de cadenas de caracteres (strings).

c) Utilizando las funciones de (a) y (b), implemente

- `function sopaDeLetras(sopa, tira)`

que dada la matriz `sopa`, y el vector `tira` (la palabra buscada), halle todas las ocurrencias de `tira` en `sopa`.

Para cada ocurrencia hallada, deberá imprimir en pantalla un mensaje apropiado :

- en qué fila y columna empieza la palabra hallada
- con qué dirección y sentido debe leerse a partir de esa posición

Ejemplo :

Al invocarse `sopaDeLetras(M, ['H', 'O', 'L', 'A'])`, siendo M la matriz del ejemplo en (a), debería imprimirse :

Ocurrencia encontrada :

comienza en

fila = 5

columna = 5

leyendo hacia arriba y a la izquierda

Aclaraciones:

- No es necesario dar un mensaje en caso de no encontrar ocurrencias.
- Si la palabra es de un sólo carácter, deberá imprimir tantos mensajes como ocurrencias tenga la palabra. Estos mensajes sólo indicarán en qué posición de la matriz fue la ocurrencia.
- Se asume que las siguientes funciones son análogas a las de la parte a) y ya están implementadas:

- `function v = obtenerVectorAbajo (M, i, j)`
- `function v = obtenerVectorDerecha (M, i, j)`
- `function v = obtenerVectorArribaALaIzquierda (M, i, j)`
- `function v = obtenerVectorAbajoALaIzquierda (M, i, j)`

Problema 2 (curso 2001)	15 ptos	
--------------------------------	----------------	--

- a) Represente el número $126 * 2^{-4}$ en punto flotante de simple precisión (un bit de signo, ocho bits de exponente y veintitrés bits de mantisa).
- b) Indique qué número real representa la siguiente codificación según la norma IEEE.

0 00010101 1101110000000000000000

- c) Represente NaN, cero e infinito en punto flotante de simple precisión.
- d) Indique en cada caso si las siguientes codificaciones en punto flotante representan un número normalizado, desnormalizado, NaN ó infinito:

i. 1 10000001 11001101100111000011101
ii. 1 00000000 10010001000000010000000
iii. 0 00000000 111111111111111111111110
iv. 0 11111110 00000000000000000000000
v. 0 00000000 0000000000000000000000001

Problema 2 (cursos anteriores a 2001)	15 ptos	2, 4, 9
--	----------------	----------------

- a) ¿Qué es el “epsilon” (e-mach) de la máquina ?
- b) Asumiendo que no existe una función que nos dé el e-mach, presente un algoritmo para calcularlo.
- c) Enumere los 5 tipos de errores causados por limitaciones de la computadora presentados en el curso; explique claramente cuándo ocurre y presente posibles soluciones en los casos que corresponda.

Problema 3	25 ptos	
-------------------	----------------	--

Se desea implementar un algoritmo de ordenación sobre un vector cualquiera de números. El funcionamiento de dicho algoritmo se ilustra en el siguiente ejemplo :

Sea $v = [3,8,6,1,5]$

Paso 1 : Considero las primeras 5 celdas de v : **[3, 8, 6, 1, 5]**

Encuentro el máximo valor de esas 5 celdas (el 8) y la posición donde se encuentra (la 2).

Luego lo intercambio con el valor que está en la posición 5, quedando el vector:

[3, 5, 6, 1, 8]

Paso 2 : Considero las primeras 4 celdas de v : **[3, 5, 6, 1, 8]**

Encuentro el máximo valor de esas 4 celdas (el 6) y la posición donde se encuentra (la 3).

Luego lo intercambio con el valor que está en la posición 4, quedando el vector:

[3, 5, 1, 6, 8]

Paso 3 : Considero las primeras 3 celdas de v : [3, 5, 1, 6, 8]

Encuentro el máximo valor de esas 3 celdas (el 5) y la posición donde se encuentra (la 2).

Luego lo intercambio con el valor que está en la posición 3, quedando el vector:

[3, 1, 5, 6, 8]

Paso 4 : Considero las primeras 2 celdas de v : [3, 1, 5, 6, 8]

Encuentro el máximo valor de esas 2 celdas (el 3) y la posición donde se encuentra (la 1).

Luego lo intercambio con el valor que está en la posición 2, quedando el vector:

[1, 3, 5, 6, 8]

Observar que luego de 4 pasos, el algoritmo ordena al vector v (de 5 celdas) de menor a mayor. En general, el algoritmo ordena a un vector de n celdas en $(n-1)$ pasos.

- a) Escriba una función en Matlab que dados un vector v y un entero $m \leq \text{length}(v)$, devuelva la posición del máximo valor que se encuentra almacenado en el vector dentro de las primeras m celdas.

Ejemplo : Si $v = [3, 5, 6, 1, 8]$ y $m = 4$ entonces la función debe devolver 2.

- b) Escriba una función en Matlab que dados un vector v y dos posiciones $i, j \leq \text{length}(v)$, devuelva el vector que resulta de intercambiar los valores almacenados en las posiciones i y j de v .

Ejemplo : Si $v = [5, 7, 2, 4, 9, 3]$, $i = 2$, $j = 5$ entonces la función debe devolver el siguiente vector [5, 9, 2, 4, 7, 3]

- c) Escriba una función en Matlab que implemente el algoritmo de ordenación descrito inicialmente. La función recibirá como parámetro el vector a ordenar, y retornará el vector ordenado según el algoritmo. Puede utilizar las funciones de las partes a) y b) aún cuando no las haya programado.