

PRIMER PARCIAL FUNDAMENTOS DE PROGRAMACIÓN

CURSO: 2019/2020

28/11/2019. 8:30. DURACIÓN MÁXIMA: 2 horas

EJERCICIO 1

Se considera un sistema informático que emplea 7 bits para codificar la mantisa (el primero reservado para el signo) y 4 bits para codificar el exponente (el primero reservado para el signo). Se empleará el convenio habitual de 0 para números positivos y 1 para números negativos. El sistema trabaja a redondeo y en coma flotante. Se pide:

Codificar en binario, en dicho sistema, el número que en base 10 es:

$\sin(\pi/4)+2=2.707106781\dots$

NOTA: Los contenidos que entraron en el examen de 2019, son diferentes a los de 2020. Este ejercicio de 2019 trata sobre el código binario (temario NO dado en 2020. Por ello, este documento no cuenta con la resolución del ejercicio 1.

EJERCICIO 2.

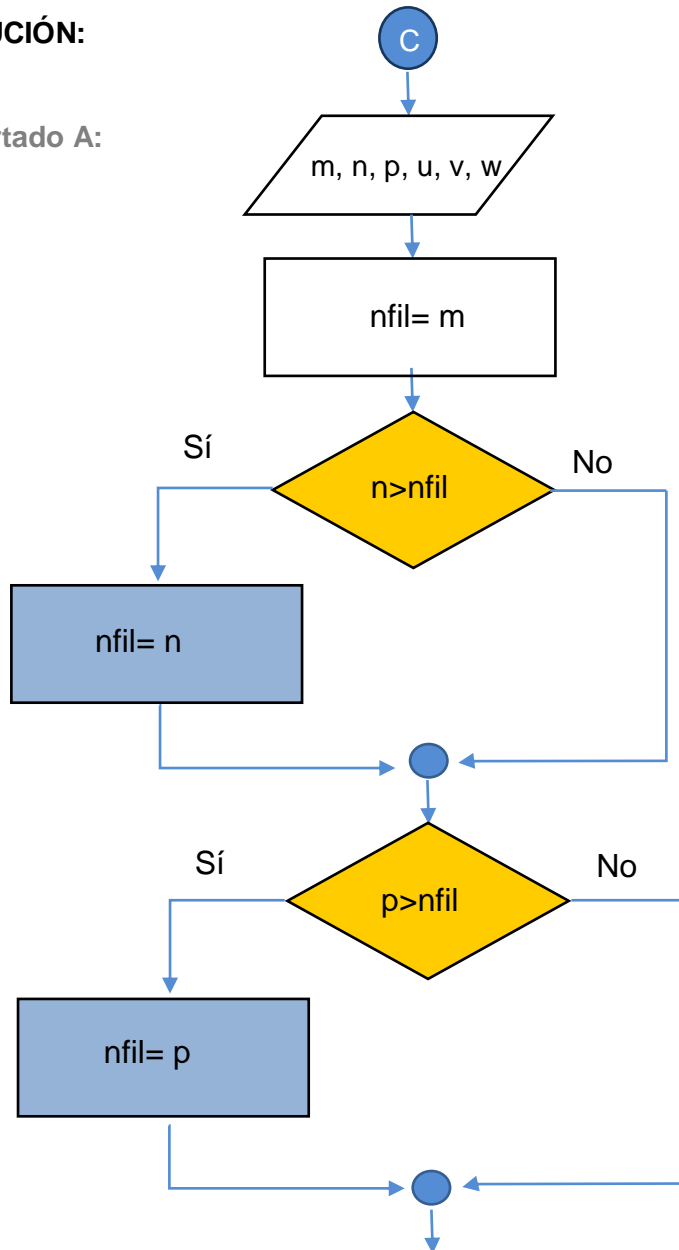
Escribir **UN ORGANIGRAMA** que realice las siguientes operaciones:

- Dados tres números enteros, m, n, p , seleccione el mayor de ellos y lo almacene en la variable $nfil$. Los datos de entrada del algoritmo incluirán también los vectores: \mathbf{u} , \mathbf{v} , \mathbf{w} ; teniendo el vector \mathbf{u} m componentes, el vector \mathbf{v} n componentes y el vector \mathbf{w} p componentes. Suponemos que estos vectores contienen valores estrictamente positivos. (0.5 puntos)
- Construya una matriz \mathbf{A} de $nfil$ filas y 3 columnas formadas por los vectores \mathbf{u} , \mathbf{v} , \mathbf{w} en este orden. Los elementos de la matriz sobrantes tendrán como valor: 0. Nota: Se recomienda inicializar $\mathbf{A}=0$ y se debe tener en cuenta que el vector \mathbf{u} ocupa m posiciones, el vector \mathbf{v} ocupa n posiciones y el vector \mathbf{w} ocupa p posiciones. (0.5 puntos)
- Intercambie la primera y la tercera fila de la matriz \mathbf{A} , y después la segunda y la tercera columnas. El resultado se almacenará en la propia matriz \mathbf{A} . (1 punto)
- Sume los elementos de las dos primeras columnas de la matriz \mathbf{A} obtenida en el apartado c) y almacene el resultado en un vector \mathbf{z} . Por lo tanto, el vector \mathbf{z} tendrá $nfil$ componentes. Es decir, la componente j de \mathbf{z} se obtendrá como la suma de los elementos de la matriz \mathbf{A} que se encuentran en la fila j y las columnas 1 y 2. (1 punto)
- Encuentre el MAYOR valor que contiene el vector \mathbf{z} (almacenándose en la variable $zmax$) y la posición que ocupa (variable $jmax$). (1 punto)

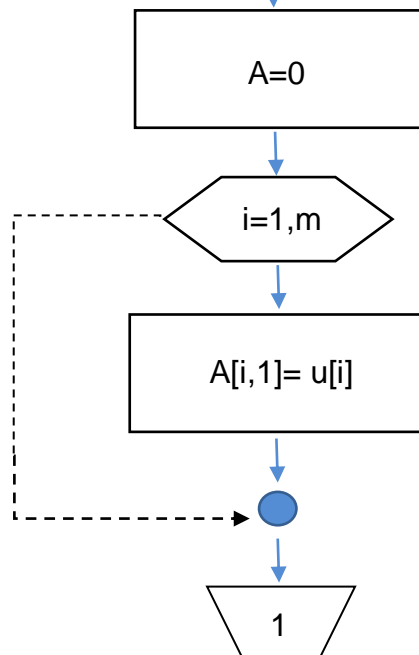
El resultado final del algoritmo será: \mathbf{A} , \mathbf{z} , $zmax$, $jmax$.

SOLUCIÓN:

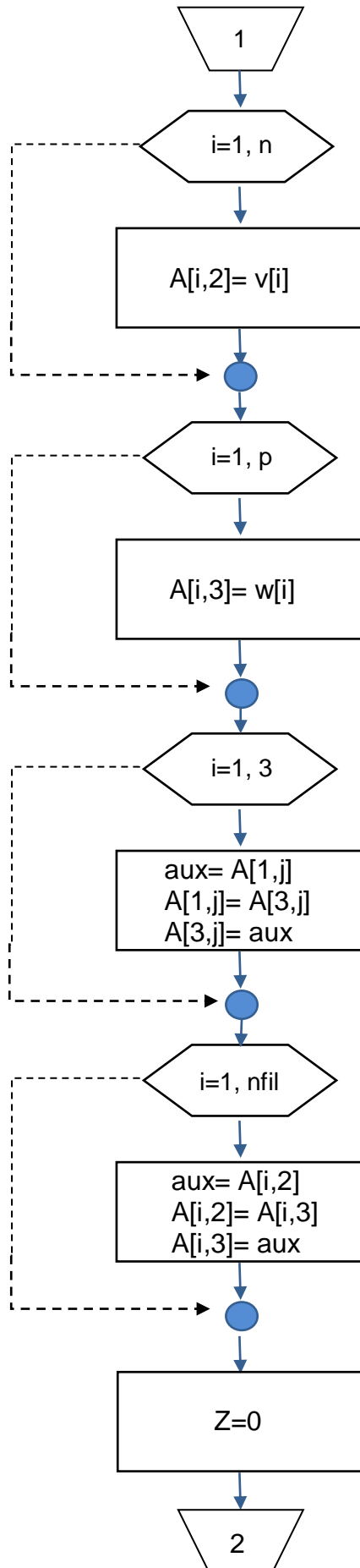
Apartado A:



Apartado B:

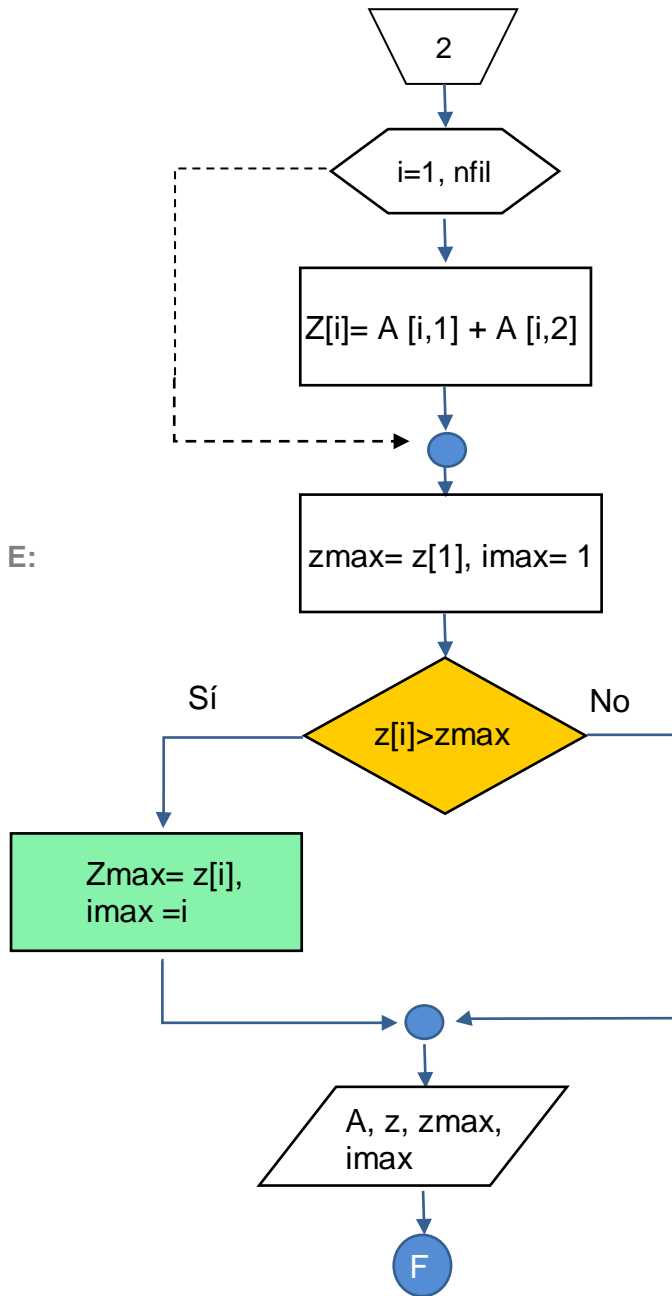


Apartado C:



Apartado D:

Apartado E:



EJERCICIO 3.

Se está realizando un cultivo de bacterias de tipo E.coli con el fin de utilizarlas en un proceso de fabricación de biocombustibles. Se ha medido la cantidad de bacterias que hay en NT instantes de tiempo (t_1, t_2, \dots, t_{NT}) conocidos, obteniendo las cantidades: B_1, B_2, \dots, B_{NT} respectivamente, también conocidas.

Se desea estimar la producción de bacterias en los instantes de tiempo: y_1, y_2, \dots, y_M conocidos y almacenar los valores obtenidos en un vector **BEST**, de M componentes. Para ello, se desea escribir **UN PSEUDO-CÓDIGO** que realice las siguientes operaciones:

- a) Obtenga una matriz, denominada **ECOLI**, cuyos elementos vengan dados por:

$$ECOLI_{i,1} = B_i \quad , (i = 1, 2, \dots, NT)$$

$$ECOLI_{i,j} = \frac{ECOLI_{i+1,j-1} - ECOLI_{i,j-1}}{t_{i+1,j-1} - t_i} \quad , (j = 2, \dots, NT; \quad i = 1, 2, \dots, NT - j + 1)$$

Los elementos de la matriz no calculados mediante las fórmulas anteriores, tendrán valor 0, por lo que se recomienda inicializar **ECOLI**=0. Téngase en cuenta que la matriz tendrá NT filas y NT columnas.

- b) Obtenga las componentes de un vector **BEST** de acuerdo con la fórmula:

$$BEST_k = ECOLI_{1,1} + \sum_{i=2}^{NT} \left(ECOLI_{1,i} \cdot \prod_{j=1}^{i-1} (y_k - t_j) \right) \quad , (k = 1, 2, \dots, M)$$

El resultado final del algoritmo será: BEST.

SOLUCIÓN:

INICIO PSEUDO-CÓDIGO

Leer t, B, NT, y, M

↑ Para i desde 1 hasta NT hacer

ECOLI_{i,1} = B_i

↓ Fin del bucle

↑ Para j desde 2 hasta NT hacer

↑ Para i desde 1 hasta NT-j+1 hacer

ECOLI_{i,j} = (ECOLI_{i+1,j-1} - ECOLI_{i,j-1}) / (t_{i+j-1} - t_i)

↓ Fin del bucle en i

↓ Fin del bucle en j

↑ Para k desde 1 hasta M hacer

BEST_k = ECOLI_{1,1}

↑ Para i desde 2 hasta NT hacer

M=1

↑ Para j desde 1 hasta i-1

hacer M=M(y_k-t_j)

BEST_k = BEST_k + ECOLI_{1,i} * M

↓ Fin del bucle en j

↓ Fin del bucle en i

↓ Fin del bucle en k

Escribir BEST

FIN PSEUDO-CÓDIGO