

# Ejercicios

# Resueltos

# Abacus

**Organización del Computador**

**A3. -** Dada una máquina Abacus con las mismas características y operaciones disponibles que el ejercicio A1 realizar un programa que resuelva lo siguiente:

En las celdas  $150_{(16)}$  y  $250_{(16)}$  se encuentran almacenados dos números X e Y. Efectuar  $X/Y$  almacenando el cociente en la celda  $300_{(16)}$  y el resto en la celda  $301_{(16)}$ .

Dir.	Contenido	Observación
4FF	0001	Constante (1)
⇒ 500	0000	Cargo 0
501	2300	Almaceno en 300
502	1250	Cargo Y
503	4000	Complemento
504	34FF	
505	3150	X - Y
506	750D	Bifurca a 50D si es menor a 0
507	2150	Almaceno en 250
508	1300	Aumento en 1 el contenido de 300
509	34FF	
50A	2300	
50B	0000	Bifurcación incondicional
50C	5502	
50D	1150	Cargo el resto de X
50E	2301	Almaceno en 301
50F	FFFF	FIN del programa

**Nota:** Se uso el método de restas sucesivas para la resolución del ejercicio. Es decir restar sucesivamente Y de X y sumar uno al cociente por cada vez que se haga, cuando ya no pueda hacerse, habremos obtenido el cociente y en donde estaba X, tendremos el resto.

**A5. -** Dada una máquina Abacus con las siguientes características:

RPI: 12 bits                      AC: 16 bits

Operaciones disponibles: (Códigos en hexadecimal)

- 1      Bifurca si el (AC) < 0
- 2      Resta
- 3      Suma
- 4      Almacena
- 5      Carga

En memoria, a partir de la celda  $230_{(16)}$  se encuentra el siguiente programa:  
0001 5233 2233 4200 5200 3201 4200 5202 3230 4202 1234

Los contenidos de las celdas  $201_{(16)}$  y  $202_{(16)}$  son  $0002_{(16)}$  y  $FFFE_{(16)}$ , respectivamente. El contenido del RPI es  $231_{(16)}$ .

**Organización del Computador**

Se pide efectuar un seguimiento del programa detallando las microinstrucciones e indicar el estado final de todos los componentes (AC, RI, RPI, RM, RDM) y las celdas que se modifican.

**Corrida del Programa**

Celdas	Contenido	Corrida del programa	Continuación de la corrida luego de bif.
230	0001		
⇒ 231	5233	(AC) = 4200	
232	2233	(AC) = (AC) - (233) = 4200 - 4200 = 0000	
233	4200	(200) = (AC) = 0000	
234	5200	(AC) = (200) = 0000	(AC) = (200) = 0002
235	3201	(AC) = (AC) + (201) = 0000 + 0002 = 0002	(AC) = (AC) + (201) = 0002 + 0002 = 0004
236	4200	(200) = (AC) = 0002	(200) = (AC) = 0004
237	5202	(AC) = (202) = FFFE	(AC) = (202) = FFFF
238	3230	(AC) = (AC) + (230) = FFFE + 0001 = FFFF	(AC) = (AC) + (230) = FFFF + 0001 = 0000
239	4202	(202) = (AC) = FFFF	(202) = (AC) = 0000
23A	1234	(AC) < 0 ==> <b>BIFURCA A 234</b>	(AC) = 0 ==> <b>NO BIFURCA</b>

AC	RI	RPI	RM	RDM
0000	1234	23B	1234	23A

200	201	202
0004	0002	0000

RPI	RDM	RM	RI		AC	200	201	202	Observaciones
			CO	OP					
231							0002	FFFE	Fase de Búsqueda
	231								RDM<-RPI
		5233							RM<-(RDM)
			5	233					RI<-RM
232									RPI<-RPI+1
	233								RDM<-OP
		4200							RM<-(RDM)
					4200				AC<-RM
233	232	2233	2	233					Fase de Búsqueda
	233								RDM<-OP
		4200							RM<-(RDM)
					0000				AC<- AC - RM
234	233	4200	4	200					Fase de Búsqueda

	200							RDM<-OP
		0000						RM<-AC
					0000			(RDM)<-RM
235	234	5200	5	200				Fase de Búsqueda
	200							RDM<-OP
		0000						RM<-(RDM)
					0000			AC<-RM
236	235	3201	3	201				Fase de Búsqueda
	201							RDM<-OP
		0002						RM<-(RDM)
					0002			AC<-AC+RM
237	236	4200	4	200				Fase de Búsqueda
	200							RDM<-OP
		0002						RM<-AC
						0002		(RDM)<-RM
238	237	5202	5	202				Fase de Búsqueda
	202							RDM<-OP
		FFFE						RM<-(RDM)
					FFFE			AC<-RM
239	238	3230	3	230				Fase de Búsqueda
	230							RDM<-OP
		0001						RM<-(RDM)
					FFFF			AC<-AC+RM
23A	239	4202	4	202				Fase de Búsqueda
	202							RDM<-OP
		FFFF						RM<-AC
							FFFF	(RDM)<-RM
23B	23A	1234	1	234				Fase de Búsqueda
234								RPI<-OP
235	234	5200	5	200				Fase de Búsqueda
	200							RDM<-OP
		0002						RM<-(RDM)
					0002			AC<-RM
236	235	3201	3	201				Fase de Búsqueda
	201							RDM<-OP
		0002						RM<-(RDM)
					0004			AC<-AC+RM
237	236	4200	4	200				Fase de Búsqueda
	200							RDM<-OP
		0004						RM<-AC
						0004		(RDM)<-RM
238	237	5202	5	202				Fase de Búsqueda
	202							RDM<-OP
		FFFF						RM<-(RDM)
					FFFF			AC<-RM
239	238	3230	3	230				Fase de Búsqueda
	230							RDM<-OP
		0001						RM<-(RDM)
					0000			AC<-AC+RM
23A	239	4202	4	202				Fase de Búsqueda

**Organización del Computador**

	202								RDM<-OP
		0000							RM<-AC
							0000		(RDM)<-RM
FIN									No hay más celdas por ejecutar

**A6. -** Dada una máquina Abacus con las mismas características y operaciones disponibles que el ejercicio A1 realizar un programa que resuelva lo siguiente:

En las celdas  $100_{(16)}$ ,  $200_{(16)}$  y  $300_{(16)}$  se encuentran almacenados tres números X,Y,Z. Se pide ordenarlos en forma ascendente y almacenarlos en ese orden a partir de la dirección  $700_{(16)}$ .

Dir.	Contenido	Observación
4FE	EAEA	Auxiliar
4FF	0001	Constante (1)
⇒ 500	1300	Z
501	2702	
502	1200	Y
503	2701	
504	1100	X
505	2700	
506	4000	Complemento
507	34FF	
508	3701	Y – X
509	6510	Si mayor que 0 => Y > X
50A	1701	Intercambio contenido de la celda 700 y el de la 701
50B	24FE	
50C	1700	
50D	2701	
50E	14FE	
50F	2700	
510	1701	Cargo el mayor entre X e Y (Mxy)
511	4000	Complemento
512	34FF	
513	3702	Z – Mxy
514	651B	Si mayor que 0 => Z > Mxy
515	1702	Intercambio contenido de la celda 701 y el de la 702
516	24FE	
517	1701	
518	2702	
519	14FE	
51A	2701	
51B	1700	Cargo el menor entre X e Y (mxy)
51C	4000	Complemento
51D	34FF	
51E	3701	m(Mxy)z – mxy
51F	6526	Si mayor que 0 => m(Mxy)z > mxy
520	1701	Intercambio contenido de la celda 700 y el de la 701
521	24FE	

## Organización del Computador

522	1700	}
523	2701	
524	14FE	
525	2700	
526	FFFF	

FIN del programa

**A8.** - Dada una máquina Abacus con las mismas características y operaciones disponibles que el ejercicio A1 realizar un programa que resuelva lo siguiente:

En la celda  $150_{(16)}$  se encuentra almacenado un número que representa la máxima longitud de un BPF c/signo en bits. Almacenar en la celda  $300_{(16)}$  el menor número almacenable en esa cantidad de bits y el mayor número almacenable en la celda  $301_{(16)}$ .

Dir.	Contenido	Observación
4FB	0001	Constante (1)
4FC	0001	Inicializo en 1. Voy guardando los resultados.
4FE	0000	Auxiliar
4FF	1111	Constante (-1)
500	1150	AC = (150)
501	34FF	AC=(AC) + (-1)
502	24FE	4FE=(AC) Celda auxiliar
503	550B	Si el exponente es 0, bifurca
504	14FC	Sumo al AC el mismo valor, $nx2=n+n$ , siendo n la acumulación de las potencias anteriores
505	34FC	
506	24FC	
507	14FE	
508	34FF	Decremento el exponente
509	24FE	
50A	6504	
50B	14FC	Guardo en las celdas los resultados
50C	34FF	
50D	2301	
50E	14FC	
50F	4FFF	Guardo en la 301 el mayor
510	34FB	
511	2300	
512	FFFF	

Guardo en el 300 el menor  
FIN del programa

**Nota:** El máximo y mínimo de un número que se representa en BPF c/signo es  $2^{n-1}-1$  y  $-2^{n-1}$  respectivamente, siendo n la cantidad de bits del formato. Primero realizamos el cálculo de  $2^{n-1}$  y luego las operaciones restantes para completar la fórmula.

**A14.** - Dada una máquina Abacus con las mismas características y operaciones disponibles que el ejercicio A11 realizar un programa que resuelva lo siguiente:

Dado un vector almacenado a partir de la celda  $200_{(16)}$  y que termina en  $-1_{(10)}$  se pide invertir al vector, es decir, el último elemento queda en el lugar del primero, (en la celda  $200_{(16)}$  queda el  $-1_{(10)}$ ), en el anteúltimo queda el segundo, etc.

Dir.	Contenido	Observación
0FB	2000	Auxiliar Almacenamiento
0FC	EAEA	NUM2
0FD	EAEA	NUM1
0FE	EAEA	Apuntará a una posición del vector (DIR2V)
0FF	FFFF	Constante (-1)
100	FFFE	Constante (-2)
101	0001	Constante (1)
102	1000	Auxiliar Carga
103	EAEA	Contador (CONT)
104	EAEA	Apuntará a una posición del vector (DIR1V)
⇒ 105	0001	Inicializo el contador en 1
106	2103	
107	0200	Almaceno la dirección de comienzo del vector
108	2104	
109	1104	Cargo el contenido de la posición del vector apuntada por DIR1V
10A	3102	
10B	210C	
10C	BACA	
10D	7116	Cuando llega al fin del vector (-1) bifurca a la celda 116
10E	1103	Sumo 1 al contador
10F	3101	
110	2103	Avanzo 1 en la posición del vector
111	1104	
112	3101	Bifurcación incondicional
113	2104	
114	0000	Almaceno la dirección de comienzo del vector en la celda 104 y la dirección de fin del vector en la celda 0FE
115	5109	
116	0200	
117	2104	
118	3103	Almaceno en NUM1 el contenido apuntado por DIR1V
119	30FF	
11A	20FE	
11B	1104	
11C	3102	Almaceno en NUM2 el contenido apuntado por DIR2V
11D	211E	
11E	BACA	Almaceno en la dirección apuntada por DIR1V el contenido de NUM2
11F	20FD	
120	10FE	
121	3102	
122	2123	Almaceno en la dirección apuntada por DIR1V el contenido de NUM2
123	BACA	
124	20FC	Almaceno en la dirección apuntada por DIR1V el contenido de NUM2
125	1104	
126	30FB	Almaceno en la dirección apuntada por DIR1V el contenido de NUM2
127	2129	
128	10FC	Almaceno en la dirección apuntada por DIR1V el contenido de NUM2
129	BACA	

**Organización del Computador**

12A	10FE	}	Almaceno en la dirección apuntada por DIR2V el contenido de NUM1
12B	30FB		
12C	212E		
12D	10FD		
12E	BACA	}	Avanzo 1 en DIR1V
12F	1104		
130	3101	}	Retrocedo 1 en DIR2V
131	2104		
132	10FE	}	Resto dos al contador
133	30FF		
134	20FE	}	Si contador <= 0 termina, de lo contrario bifurca a 11B
135	1103		
136	3100	}	FIN del programa
137	2103		
138	611B		
139	FFFF		

**A15.** - Dada una máquina Abacus con las siguientes características:

CO = 4 bits

OP = 12 bits

Operaciones disponibles: (Códigos en hexadecimal)

- |                    |                      |
|--------------------|----------------------|
| 0: Carga inmediata | 4: Suma 1 al AC.     |
| 1: Carga.          | 5: Bifurca si AC < 0 |
| 2: Almacenamiento. | 6: Resta             |
| 3: Suma            | 7: Bifurca si AC = 0 |
|                    | 8: Fin de programa.  |

Sabiendo que en la dirección  $1A0_{(16)}$  se encuentra la dirección de inicio de un vector, cuyo fin está indicado con una celda con valor negativo, se pide realizar un programa Abacus que ubique al principio del vector todos los elementos distintos de cero y al final del mismo los iguales a cero (debe mantener en su lugar la marca de fin de vector).

El punto de carga del programa será la celda  $200_{(16)}$ .

Ejemplo:

Si tenemos el siguiente vector:

4	0	0	1	7	9	0	0	6	-1
---	---	---	---	---	---	---	---	---	----

Debe quedar:

4	1	7	9	6	0	0	0	0	-1
---	---	---	---	---	---	---	---	---	----

Celdas del vector

4	...	-1
---	-----	----

Dir X

Celda puntero

Dir X
-------

$1A0$



**Nota:**

Para la resolución del problema se usarán dos celdas auxiliares como punteros a las celdas del vector. El puntero de la celda 105 contendrá en cada momento la dirección de la celda en la cual se puede insertar un nuevo número distinto de cero. Por otro lado el puntero de la celda 106 contendrá la dirección de la celda del vector a ser analizada en cada momento.

Se recorrerán todas las celdas del vector y en caso de encontrar en una celda un número distinto de cero se moverá este a la celda apuntada por 105 y se pondrá en cero la posición apuntada por 106 de forma tal de que al terminar la recorrida del vector queden todos los números distintos de cero al inicio del vector y las demás celdas en cero.

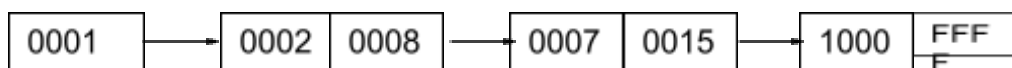
Dir.	Contenido	Observación
100	1000	Celda auxiliar para "armar" una instrucción de carga
101	2000	Celda auxiliar para "armar" una instrucción de almacenamiento
105	Dir celda	Puntero a la próxima posición de inserción en el vector
106	Dir celda	Puntero a la próxima celda del vector a analizar
...	...	
200	11A0	
201	2105	Inicializo punteros
202	2106	
203	3100	
204	2205	Armo instrucción de carga de celda del vector
205	CAFE	Instrucción de carga de cada celda del vector. ( 1 <i>dir.celda</i> )
206	5222	¿Es fin del vector?
207	721B	¿La celda actual del vector tiene un cero?
208	1106	
209	6105	Resta de punteros
20A	7218	¿Los punteros apuntan a la misma celda del vector?
20B	1106	
20C	3100	Muevo celdas del vector. Armo carga de celda del vector
20D	2211	
20E	1105	
20F	3101	Muevo celdas del vector. Armo almacenamiento de celda
210	2212	
211	ABAA	Instrucción de carga de celda del vector. ( 1 <i>dir.celda</i> )
212	ABAB	Instrucción de almacenamiento en celda del vector. ( 2 <i>dir.celda</i> )
213	1106	
214	3101	
215	2217	Poner un cero en la celda actual del vector
216	0000	
217	ABAD	Instrucción de almacenamiento en celda del vector. ( 2 <i>dir.celda</i> )
218	1105	
219	4000	Incrementar puntero de próxima celda de inserción.
21A	2105	
21B	1106	
21C	4000	Incrementar puntero de próxima celda del vector.
21D	2106	
21E	3100	
21F	2205	
220	0000	Volver al ciclo de recorrida del vector
221	7205	
222	8000	Fin del programa.

**A16.** - Dada una máquina Abacus con las mismas características y operaciones disponibles que el ejercicio A11 realizar un programa que resuelva lo siguiente:

Se sabe que en la dirección  $3A0_{(16)}$  se encuentra la dirección de inicio del primer nodo de una lista cuyos nodos están formados por dos celdas contiguas. La primera celda de cada nodo contiene un valor en BPF c/signo de 16 bits y la segunda contiene la dirección de inicio del nodo siguiente. En el último nodo la segunda celda contiene un  $-1_{(10)}$  para indicar el fin de la lista. Se pide recorrer todos los nodos de la lista y almacenar la sumatoria de todos los BPF c/signo de 16bits. Dicha sumatoria se deberá almacenar en la celda  $400_{(16)}$ .

Dir.	Contenido	Observación
102	FFFF	Constante (-1)
103	0001	Constante (1)
104	1000	Auxiliar Carga
⇒ 105	13A0	$(AC) = (3A0) = 0001$
106	3104	$(AC) = (3A0) + (104) = 0001 + 1000 = 1001$
107	2108	$(108) = (AC) = 1001$
108	FAFA	$(AC) = ((3A0))$
109	3400	$(AC) = (AC) + (400) = 0002 + 0000$
10A	2400	$(400) = 0002$ Acumulo
10B	1108	$(AC) = 1(108) = 1001$
10C	3103	$(AC) = (AC) + (103) = 1001 + 0001 = 1002$
10D	210E	$(10E) = (AC) = 1002$ nuevo el puntero
10E	FAFA	$(AC) = (002) = 0008$
10F	3103	$(AC) = (AC) + (103) = 0008 + 0001 = 0009$
110	5116	<b>SI (AC) = 0 BIFURCA A FIN DE PROG.</b>
111	3102	$(AC) = (AC) + (102) = 0009 + FFFF = 0008$
112	3104	$(AC) = (AC) + (104) = 0008 + 1000 = 1008$
113	2108	$(108) = (AC) = 1008$
114	0001	$(AC) = 0001$
115	6108	<b>SI (AC) &gt; 0 BIFURCA A CELDA 108</b>
116	FFFF	FIN del programa

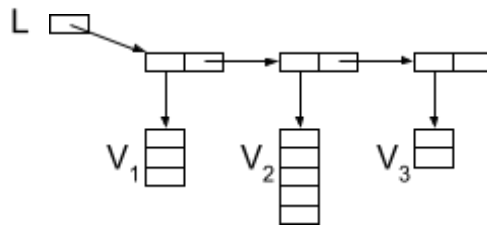
Lista para hacer el seguimiento:



**A19.** - Dada una máquina Abacus con las mismas características y operaciones disponibles que el ejercicio A11 realizar un programa que resuelva lo siguiente:

Se tiene una lista (L) en donde cada nodo contiene una celda que apunta al inicio de un vector de celdas (V) y otra celda que contiene la dirección del siguiente nodo de la lista. El vector de celdas contiene en cada celda un BPF sin signo de 12 bits y termina con un valor 0. A su vez el final de la lista está determinado por un valor  $-1_{(10)}$  en la segunda celda del último nodo. Se pide realizar un programa Abacus que recorra todas las celdas de los vectores (V)

de todos los nodos de la lista (L) y obtenga el máximo BPF sin signo almacenado. Este valor se deberá dejar almacenado en la celda 200<sub>(16)</sub>. La celda 300<sub>(16)</sub> contiene la dirección de inicio de L.



Dir.	Contenido	Observación
101	0001	Constante (1)
102	EAEA	NUM
103	EAEA	Puntero para recorrer el vector (DIRV1)
104	1000	Auxiliar Carga
⇒ 105	0000	Inicializo la celda 200
106	2200	
107	1300	
108	7125	Si es -1: FIN DE LISTA
109	3104	
10A	210B	Cargo el contenido de la celda apuntada por la celda 300 y lo almaceno en la 103 (Puntero al vector)
10B	AAAA	
10C	2103	
10D	1103	
10E	3104	Cargo el contenido de la celda apuntada por la celda 103
10F	2110	
110	AAAA	
111	511E	
112	2102	Si es 0: FIN DE VECTOR
113	4000	Almaceno el valor en NUM
114	3101	NOT NUM
115	3200	- NUM
116	6119	MAX - NUM
117	1102	Si es mayor a cero (MAX > NUM), no modifica el valor de MAX
118	2200	Cambio el valor de MAX (celda 200) por el valor de NUM
119	1103	
11A	3101	Avanza 1 el puntero del vector
11B	2103	
11C	0000	Bifurcación incondicional
11D	510D	
11E	110B	Hace a (300) apuntar al siguiente nodo de la lista
11F	3101	
120	2121	
121	AAAA	
122	2300	Bifurcación incondicional
123	0000	
124	5107	FIN del programa
125	FFFF	

**A20.** – Dada una máquina Abacus con las siguientes características:

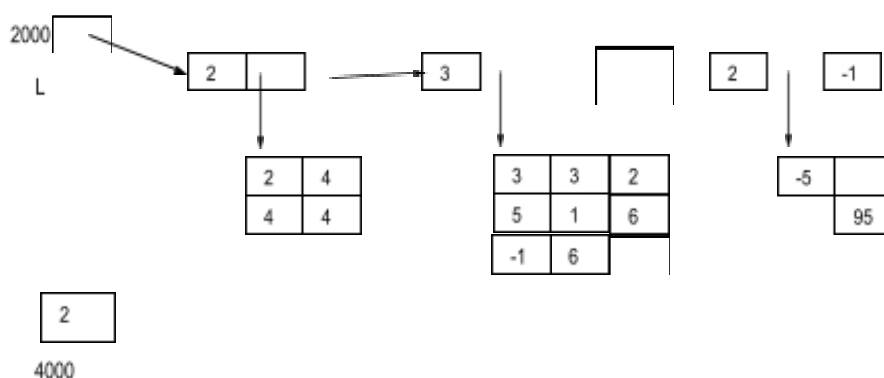
RPI = 12 bits AC = 16 bits

Operaciones disponibles: (Códigos en hexadecimal)

0	Carga inmediata	7	Bifurcación si (AC) = 0
1	Cargar	8	Bifurcación si (AC) > 0
2	Almacenar	9	Bifurcación si (AC) < 0
3	Sumar	F	Fin de Programa
6	Not (AC)		

Se tiene una lista (L) cuya dirección de inicio se encuentra almacenada en la celda 200<sub>(16)</sub>. Cada nodo de la lista (L) está formado por tres celdas contiguas en memoria. La primera celda contiene un número n que representa la dimensión de una matriz cuadrada de n\*n. La segunda celda contiene la dirección de inicio de una matriz (M) cuadrada. La tercera celda del nodo contiene la dirección del siguiente nodo de la lista. El final de la lista (L) se indica con un valor -1 en la tercera celda del último nodo.

Se pide realizar un programa ABACUS con punto de carga en la celda 300<sub>(16)</sub> que recorra todos los nodos de la Lista (L) y calcule la sumatoria de los elementos de la diagonal principal de cada matriz (M) asociada almacenando en la celda 400<sub>(16)</sub> la mínima sumatoria calculada.



### Dir. Contenido Observación

2F9	FFFF	Auxiliar -1
2FA	0000	Contador de filas de matriz
2FB	0000	Flag de sumatoria
2FC	0000	Sumatoria
2FD	0001	Auxiliar 1
2FE	FFFF	Dimensión de la matriz cuadrada
2FF	1000	Auxiliar Carga
→ 300	0000	Inicializo la celda 200
300	1200	
301	32FF	Genero carga de dimensión
302	2307	Almaceno en 307
303	32FD	Genero carga de inicio matriz

## Organización del Computador

304	2309	Almaceno en 309
305	32FD	Genero carga de dir de siguiente nodo
306	232F	Almaceno en 32F
307	FFFF	Carga de dimensión
308	22FE	Almaceno la dimensión
309	FFFF	Carga de inicio matriz
30A	32FF	Genero carga de celda matriz
30B	230C	Almaceno en 30C
30C	FFFF	Cargo celda matriz
30D	32FC	Sumo el valor de sumatoria
30E	22FC	Almaceno el resultado
30F	12FA	Cargo contador de Filas
310	32FD	Incremento contador de filas
311	22FA	Guardo contador de filas
312	6FFF	Complemento
313	32FD	Dimensión - Contador de Filas
314	32FE	Bifurco si terminé sumatoria
315	731C	
316	130C	Cargo dir. De última celda cargada
317	32FE	Sumo dimensión de matriz
318	32FD	Sumo 1
319	230C	Almaceno dir. de próxima celda
31A	0000	Bifurcación Incondicional
31B	730C	Cargo Flag de sumatoria
31C	12FB	
31D	8324	Si no es la primer sumatoria
31E	32FD	Cambio estado de Flag
31F	22FB	
320	12FC	Cargo resultado de sumatoria
321	2400	Almaceno en 400
322	0000	Bifurcación Incondicional
323	732C	
324	12FC	Cargo sumatoria actual
325	6FFF	Complemento
326	32FD	
327	3400	Sumatoria Menor – Sumatoria Actual
328	932C	Bifurco si no es menor
329	732C	
32A	12FC	Actualizo Sumatoria Menor
32B	2400	
32C	0000	Cargo 0
32D	22FA	Limpio Contador de Filas de Matriz
32E	22FC	Limpio Auxiliar de Sumatoria
32F	FFFF	Cargo dir. de siguiente nodo
330	9334	Si es -1 bifurco a fin de programa
331	2200	Almaceno la dirección en la celda 200
332	0000	Bifurcación Incondicional
333	7300	
334	FFFF	Fin de Programa