



UNAH-CURC
CENTRO UNIVERSITARIO REGIONAL
DEL CENTRO

Universidad Nacional Autónoma De Honduras Centro Universitario Regional Del Centro

Asignatura:

Programación II

Catedrático:

Ing. Elmer Padilla

Alumno:

Angel Moises Martinez Mejia

Sección:

1100

Fecha:

19/06/2023

- a) Usar un ciclo controlado por centinela para leer 10 números positivos. Calcular e imprimir la suma.

Ubicación	Código
00	+1015
01	+2015
02	+4004
03	+4300
04	+1016
05	+2016
06	+4009
07	+1115
08	+4300
09	+3015
10	+2115
11	+1016
12	+4009
13	+1115
14	+4300
15	+0000
16	+0000

- 1) Recibe un número y para si es negativo (00 a 03) Si el número es positivo, se guarda en la variable Suma.
- 2) Recibe un segundo número y si es negativo, se imprime Suma y se para (04-08) Si el número es positivo, se suma a la variable Suma.
- 3) Se realiza el ciclo hasta que se recibe un número negativo (09-12) 4) Se imprime la suma y se detiene la ejecución (13-14) 5) Variables (15 y 16)

- b) Usar un ciclo controlado por contador para leer siete números, algunos positivos y otros negativos, y calcular e imprimir su promedio

Ubicación	Código
00	+1020
01	+2020
02	+4100
03	+2020
04	+4200
05	+1024
06	+2024
07	+3034
08	+2123

09	+2121
10	+2022
11	+2121
12	+2021
13	+3120
14	+4005
15	+2021
16	+3223
17	+2125
18	+1125
19	+4300
20	+0000
21	+0000
22	+0000
23	+0000
24	+0000

No recibe 7 números, sino cualquiera que el usuario introduzca al principio. Por lo demás, el procedimiento es igual para 7 o para cualquier número N. Se requiere una variable que haga las veces de contador, en la localidad 21, inicializada en 0. Esta variable se incrementa cada vez que se realiza el ciclo. El incremento es siempre en uno, y se hace sumando la constante uno almacenada en la localidad 22. También es necesaria otra variable más, llamada suma, localizada en la localidad 23, la cual se inicia con el valor 0 y se incrementa cada vez con el valor del número recibido. El programa consta de 6 partes, separadas por espacios para hacer más visible esta división. Éstas partes son:

1) Recepción del límite (de la localidad 00 a la 04) Verifica que el primer número (el que indica cuántos enteros se recibirán) sea positivo. Para esto, primero se revisa que no sea negativo; si lo es, el control se regresa al inicio, a la localidad 00. Después de esto, se verifica que el número sea distinto de 0. Si lo es, de nuevo se regresa el control a la localidad 00.

2) Suma del número. (De las localidades 05 a 08) Recibe un número y lo suma al contenido de la variable suma, la cual, inicialmente tiene el valor 0.

3) Incremento del contador (De las localidades 09 a 11) Para incrementar la variable Contador, almacenada en la localidad 21, se hace uso de la constante Uno, almacenada en la localidad 22.

4) Verificación de la condición para verificar que Contador no haya superado a Límite, se resta el primero del segundo. En caso de que el resultado no sea positivo (esto incluye 0 o un número negativo), se termina el ciclo, de lo contrario, se transfiere el control a 05.

5) Impresión de los resultados (de la localidad 15 a la 19) En estas instrucciones se divide la Suma entre el Límite y se presenta el resultado.

6) Variables (Localidades 20 a 25) En esta sección se almacenan las variables a las que se hace referencia en el programa.

- c) Leer una serie de números, determinar e imprimir el número más grande. El primer número leído indica cuántos números deben procesarse.

Ubicación	Código
00	+1026
01	+2026
02	+4100
03	+2026
04	+4200
05	+1024
06	+2025
07	+3126
08	+4221
09	+1023
10	+2024
11	+3123
12	+4015
13	+2023
14	+2124
15	+2025
16	+3027
17	+2125
18	+2025
19	+3126
20	+4109
21	+1124
22	+4300
23	+0000
24	+0000

25	+0000
26	+0000
27	+0000

También éste programa ha sido dividido en partes para su mejor comprensión.

- 1) Recepción del límite (00 a 04) Ésta variable indica cuántos números se recibirán. No se aceptan negativos ni 0.
- 2) Recepción del primer número (05) Éste número, por ser el primero, se almacena en la variable Mayor, localidad 24.
- 3) Primera verificación del límite (06-08) En caso de que el límite sea sólo 1 número, éste condicional lleva el control a la localidad 21.
- 4) Recepción de número (09) Si el límite es 2 o más, se recibe el siguiente número y se almacena en la variable Número (localidad 23).
- 5) ¿Es más grande Mayor o Número? (10 a 12) Si Mayor - Número > 0, entonces el control se transfiere a la localidad 15, de lo contrario se lleva a cabo las instrucciones siguientes.
- 6) Mayor = Número (13-14) Estas líneas asignan un nuevo valor a la variable Mayor.
- 7) Incremento del contador (15-17) Éste incremento hace uso de la variable Uno, la cual se inicializa en 1.
- 8) Verificación del contador (18- 20) Si Contador - limite es negativo, se transfiere el control a la localidad 9. De lo contrario (una diferencia igual a 0), se realiza la siguiente instrucción.
- 9) Se imprime el número Mayor y se termina el programa (21-22) Esta instrucción se realiza cualesquiera que sean las transferencias de control anteriores.
- 10) Variables (23-27) La variable Número almacena los sucesivos números que se introducen. La variable Mayor almacena siempre el mayor, Contador debe ser inicializado a 0, Límite se recibe del usuario al principio, y Uno debe inicializarse a 1.

Código de operación

Significado

Operaciones de entrada/salida:

`final int LEE = 10;`

Lee una palabra desde el teclado y la introduce en una ubicación específica de memoria.

`final int ESCRIBE = 11;`

Escribe una palabra de una ubicación específica de memoria y la imprime en la pantalla.

Operaciones de carga/almacenamiento:

`final int CARGA = 20;`

Carga una palabra de una ubicación específica de memoria y la coloca en el acumulador.

`final int ALMACENA = 21;`

Almacena una palabra del acumulador dentro de una ubicación específica de memoria.

Operaciones aritméticas:

`final int SUMA = 30;`

Suma una palabra de una ubicación específica de memoria a la palabra en el acumulador (deja el resultado en el acumulador).

`final int RESTA = 31;`

Resta una palabra de una ubicación específica de memoria a la palabra en el acumulador (deja el resultado en el acumulador).

`final int DIVIDE = 32;`

Divide una palabra de una ubicación específica de memoria entre la palabra en el acumulador (deja el resultado en el acumulador).

`final int MULTIPLICA = 33;`

Multiplica una palabra de una ubicación específica de memoria por la palabra en el acumulador (deja el resultado en el acumulador).

Operaciones de transferencia de control:

`final int BIFURCA = 40;`

Bifurca hacia una ubicación específica de memoria.

`final int BIFURCANEG = 41;`

Bifurca hacia una ubicación específica de memoria si el acumulador es negativo.

`final int BIFURCACERO = 42;`

Bifurca hacia una ubicación específica de memoria si el acumulador es cero.

`final int ALTO = 43;`

Alto. El programa completó su tarea.