



Instituto Tecnológico y de Estudios Superiores de Monterrey

Escuela de Ingeniería y Ciencias

## **Modelación de Sistemas Mínimos y Arquitecturas Computacionales**

Examen Argumentativo

Profesor:

Emmanuel Torres Rios

Grupo:

101

Estudiantes:

ICT - Rodrigo López Guerra A01737437

7 de Septiembre del 2023

## Programa realizado

/ Programa realizado por Rodrigo López Guerra  
/ A01737437

Loop, Load result  
Clear  
Store result  
Input  
Output  
Store x  
Load counter  
Add x  
Store counter  
Load counter2  
Add seven  
Store counter2

XSQUARE, Load result  
Add x  
Store result  
Load counter  
Subt one  
Store counter  
Load counter  
Skipcond 800  
jump SEVENX  
jump XSQUARE

SEVENX, Load result  
Add x  
Store result  
Load counter2  
Subt one  
Store counter2  
Skipcond 800  
jump TWELVEADD  
jump SEVENX

TWELVEADD, Load result  
Add twelve  
Store result  
Load result

Output  
Load case  
subt one  
Store case  
Skipcond 800  
Halt  
Jump Loop

result, DEC 0  
x, DEC 0  
counter, DEC 0  
counter2, DEC 0  
seven, DEC 7  
one, DEC 1  
twelve, DEC 12  
case, DEC 5

## Explicación del programa

El examen requiere un programa que pueda resolver la ecuación  $x^2 + 7x + 12$ .

La solución por la cual se optó fue la realización de bloques de código que ayudarán a que el programa corriera de manera correcta.

El programa acepta 5 casos de prueba para poder demostrar que todos los casos corren de manera correcta. Lo que se realiza básicamente son varios loops. El primer loop son de los 5 casos prueba, reseteando los valores de “result” “counter” “counter2” y “x”. Este se repetirá “case” veces.

El segundo loop nos realizará el cuadrado de nuestro valor “x”. Lo que se realizará es la programación de “counter”, la cual tendrá el valor de “x”, para así guardar el resultado de la operación en la variable “result”, resultando así en la variable “result” más la variable “x” la cantidad de “counter” veces, así sacando la parte de  $x^2$ .

El tercer loop nos realizará la multiplicación de 7 por “x”. Lo que se realizará es que “counter2” tendrá siempre el valor de 7, por lo cual tendremos el previo resultado de “result” más la variable “x” la cantidad de “counter2” (7) veces, así sacando la parte de  $7x$ .

Para finalizar se tiene el caso de la presentación de resultados, el cual solamente va a sumar la variable “twelve” a nuestro código y presentar el output.

## Lectura del programa

Al ser una calculadora para probar 5 casos prueba, se optó por poner en el output el valor de X, y el valor de su resultado, leyéndolo de la siguiente manera

3 ← Valor de X

42 ← Valor del resultado de la ecuación

## Captura de Pantalla de 5 Casos Pruebas

Se probó el código 5 veces con los valores de “3”, “7”, “10”, “20”, y “24”. En caso de que no se entienda la lectura de los resultados de output, se pueden referenciar de la parte de “Lectura del Programa” de este documento.

Assembly code: Autosaved file

```
1 / Programa realizado por Rodrigo López Guerra
2 / A01737437
3
4 Loop, Load result
5 Clear
6 Store result
7 Input
8 Output
9 Store x
10 Load counter
11 Add x
12 Store counter
13 Load counter2
14 Add seven
15 Store counter2
16
17 XSQUARE, Load result
18 Add x
```

Machine halted normally.

AC 0000  
IR 7000  
MAR 028  
MBR 7000  
PC 029  
IN 0018  
OUT 02F4

OUTPUT MODE: DEC

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
000	102A	A000	202A	5000	6000	202B	102C	302B	202C	102D	302E	202D	102A	302B	202A	102C
010	402F	202C	102C	8800	9016	900C	102A	302B	202A	102D	402F	202D	8800	901F	9016	102A
020	3030	202A	102A	6000	1031	402F	2031	8800	7000	02F4	0018	0000	0000	0007	0001	
030	000C	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
040	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
050	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
060	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
070	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
080	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
090	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0A0	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0B0	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

## Lectura de Output

3 ← Valor de X actual

42 ← Valor del resultado de la ecuación cuando “x” vale 3

7 ← Valor de X actual

110 ← Valor del resultado de la ecuación cuando “x” vale 7

10 ← Valor de X actual

182 ← Valor del resultado de la ecuación cuando “x” vale 10

20 ← Valor de X actual

552 ← Valor del resultado de la ecuación cuando “x” vale 20

24 ← Valor de X actual

42 ← Valor del resultado de la ecuación cuando “x” vale 24

## ¿Tiene límite nuestro código?

Sí, cuando se pone el número **178** como valor de “x”, el programa no nos va a responder, gracias a que MARIE.JS no puede manejar valores mayores a 32767, gracias a que esto puede alentar a nuestro procesador. Este valor nos va a arrojar un **OverFlow Error**, como se muestra en la siguiente captura de pantalla.

Assembly code: Autosaved file

```
41 Load result
42 Output
43 Load case
44 sub1 one
45 Store case
46 Skipcond 800
47 Halt
48 Jump Loop
49
50
51 result, DEC 0
52 x, DEC 0
53 counter, DEC 0
54 counter2, DEC 0
55 seven, DEC 7
56 one, DEC 1
57 twelve, DEC 12
58 case, DEC 5
```

AC 7FF0  
IR 302B  
MAR 02B  
MBR 00B2  
PC 018  
IN 00B2  
OUT 00B2

OUTPUT MODE: DEC 178

Overflow Error: the value 32930 is beyond the calculable range:

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
000	102A	A000	202A	5000	6000	202B	102C	302B	202C	102D	302E	202D	102A	302B	202A	102C
010	402F	202C	102C	8800	9016	900C	102A	302B	202C	102D	402F	202D	8800	901F	9016	102A
020	3030	202A	102A	6000	1031	402F	2031	8800	7000	9000	7FF0	00B2	0000	0001	0007	0001
030	000C	0005	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
040	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
050	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
060	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
070	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
080	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
090	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0A0	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0B0	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

## Lectura de Output

178 ← Valor de X actual

(No se arroja un resultado de la ecuación gracias a que hay un OverFlow Error antes de llegar al Halt).

## **Nota al Profesor**

Por alguna razón Canva no me deja subir el .txt, por lo cual le dejo una liga a Google Drive con el .txt y se lo incluyo en el documento.

<https://drive.google.com/drive/folders/1TXrEBdK2K-MzueOsJmK-5qriMLZmEygo?usp=sharing>