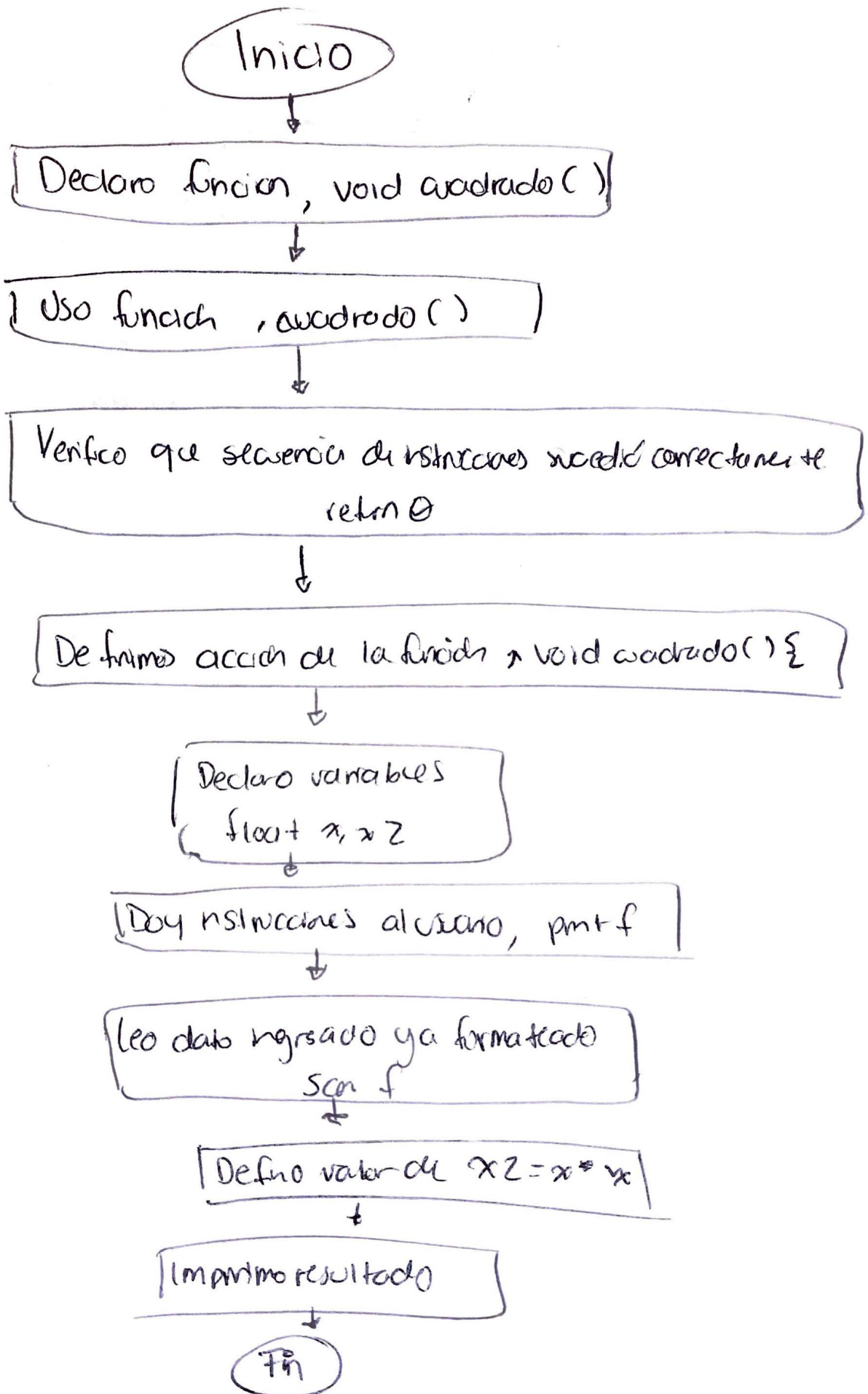
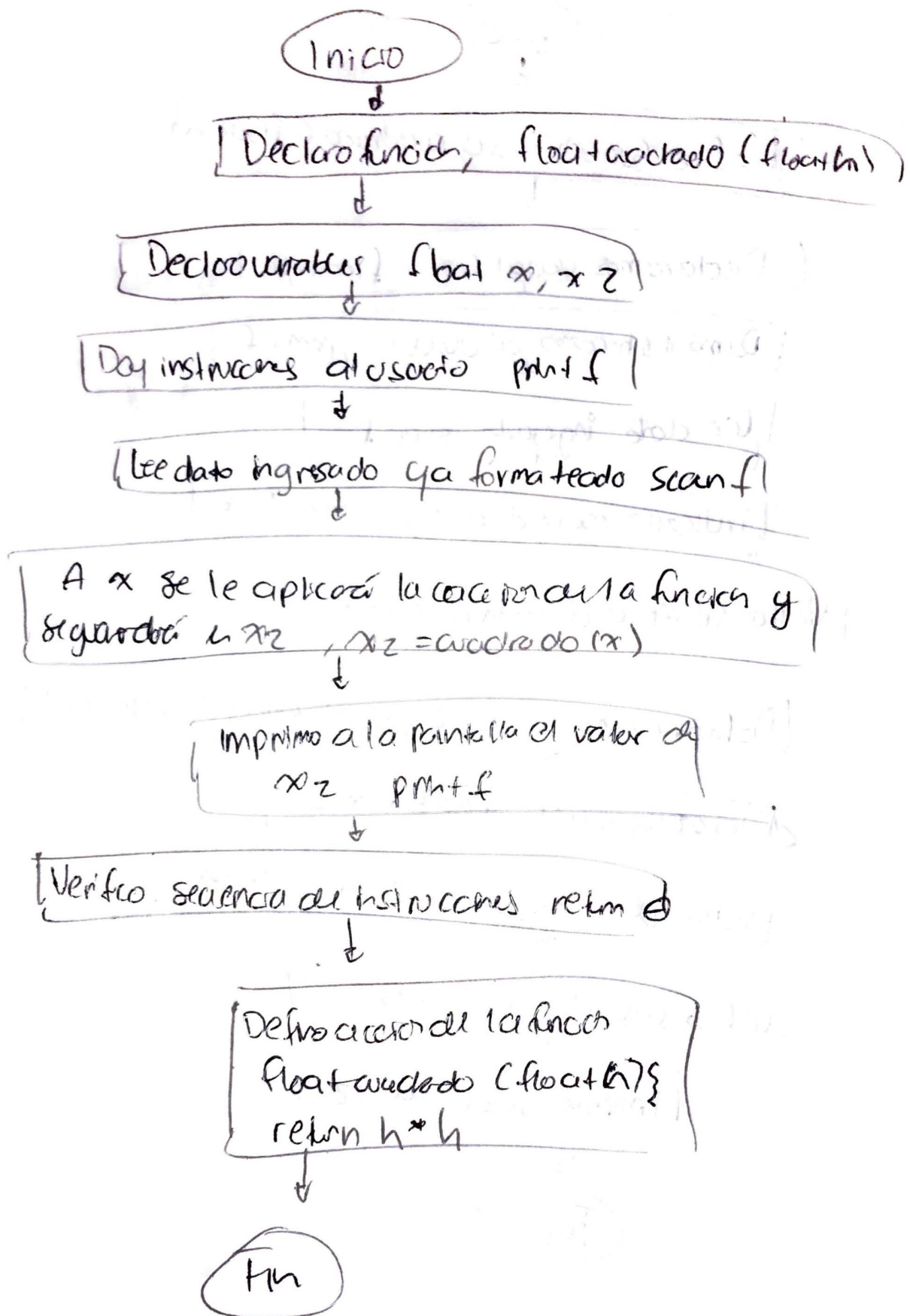


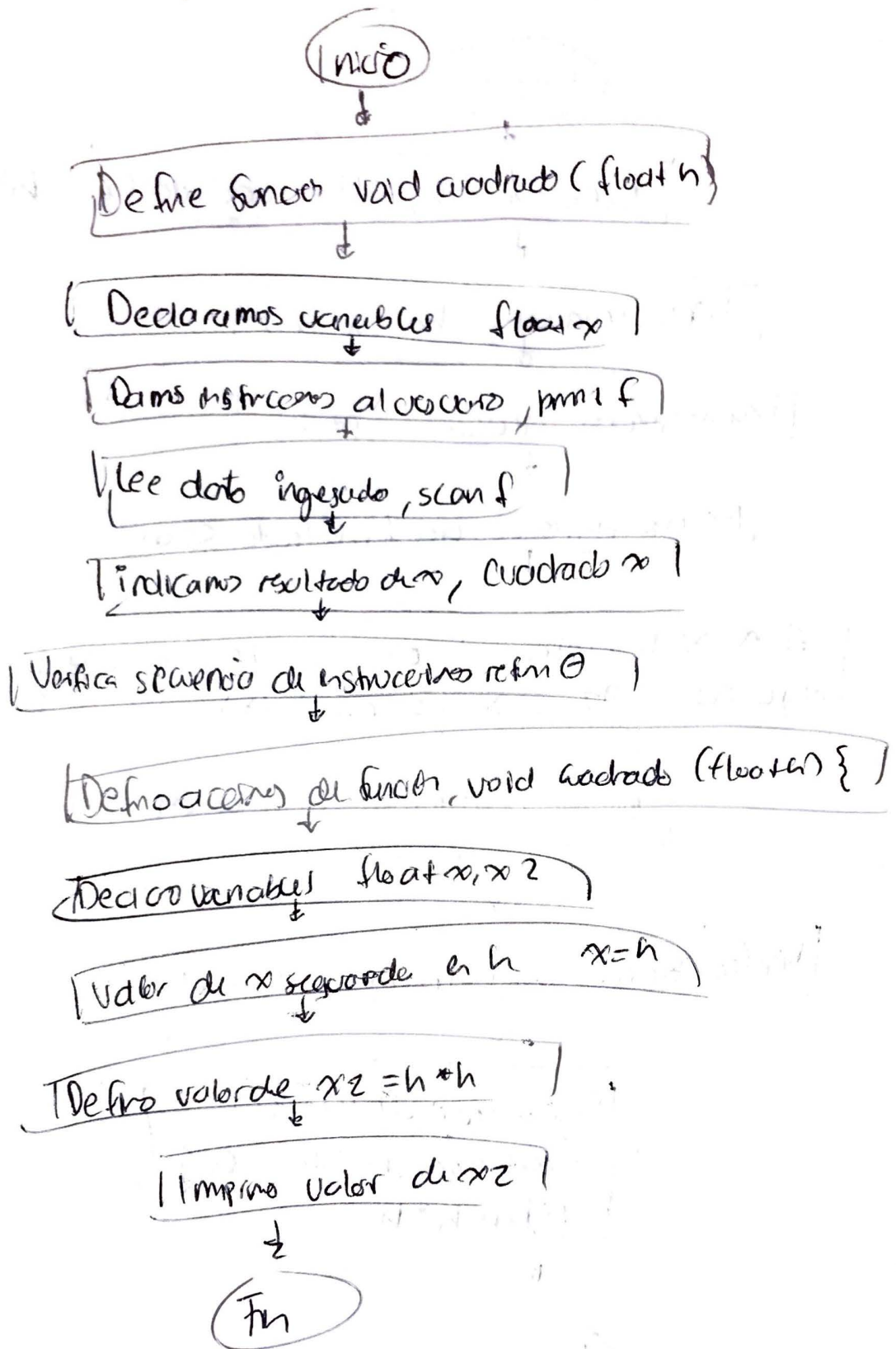
Semana 9 // ejemplo 1-c // Guadalupe Florien



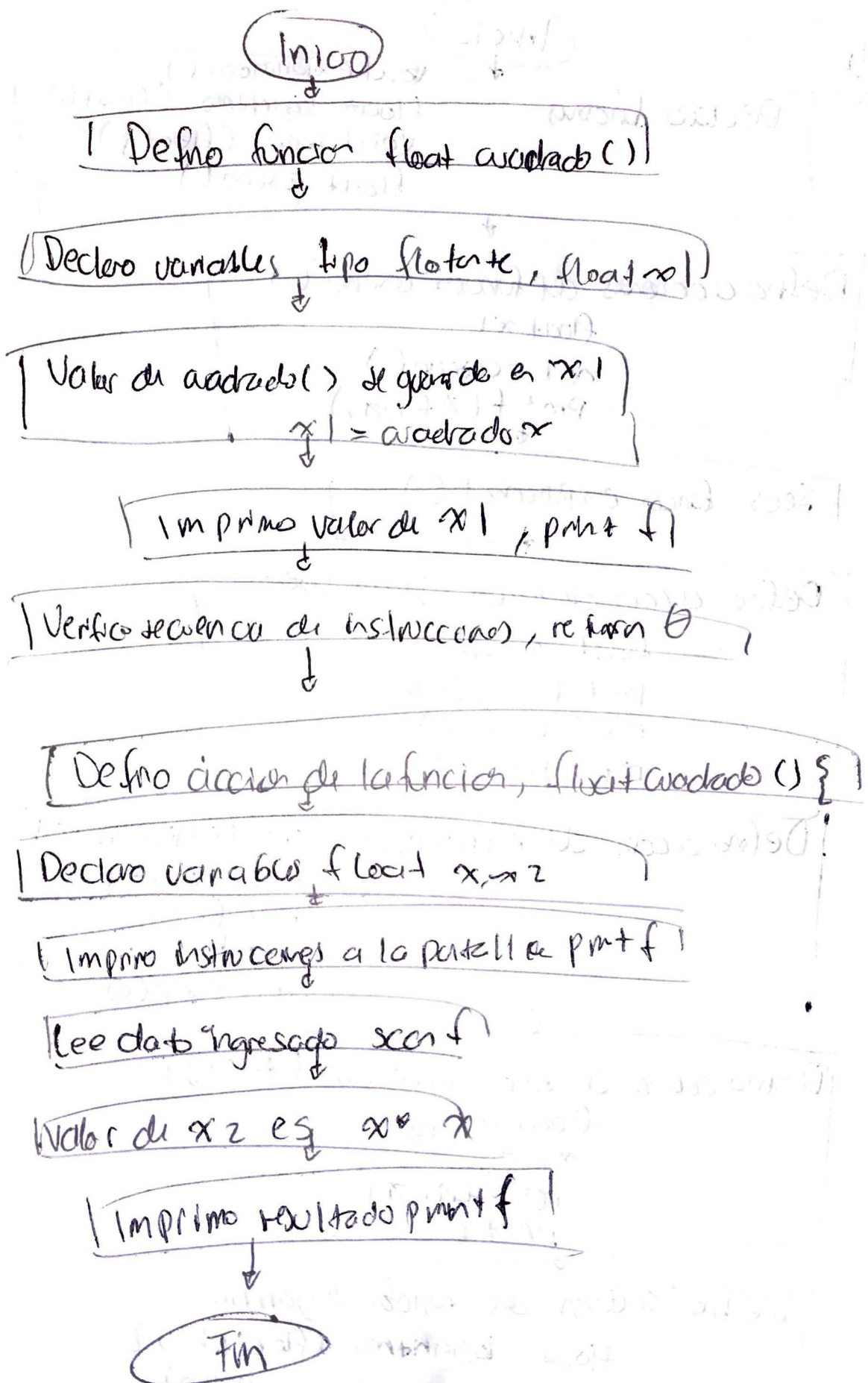
ejemplo 2.c // Semra 9 // Guadalupe Florica



ejemplo 3-c // Semana 9 // Gradualy Flow



Ejemplo 4 // Semana 9 // Guadalupe Flores



funcions.c // Guadalupe Floron // Semera 9

Inicio

Declaro funcions

```
void exponencial()
float logaritmo (float h)
void seno (float j)
float coseno()
```

Defino acciones de funci3n coseno()

```
float x1
x1 = coseno()
pmt + f(xf, x1)
```

Se usa funci3n exponencial()

Defino acci3n de funci3n float coseno

```
float x0, res
pmt + f // sen f
res = cos(x0)
return res - cos
```

Defino acci3n de exponencial

```
void exponencial()
float x1, res
pmt + f
sen f
res = exp(x1)
```

Defino acci3n de seno void seno (float j)

```
float x1, res
x1 = j
res = sen(x1)
pmt + f
```

Defino acci3n de funci3n logaritmo

```
float logaritmo (float h) {
return lg(h)
```

Fin

Semana 9 // nom-maximo.c // Geordalipe Flores

Inicio

Declaro variable archivo FILE * archivo

Declaro variables enteros y punto float a te
int i, c, count = 0;
* ptr, nom-max;

Abro archivo archivo = open (a, +xt)

Verificamos si archivo tiene contenido
if (archivo == NULL)
printf (N/A)
return 0

Calculo caracter del archivo for (c = getc (archivo))

Incremento contador si caracto empieza en una linea
if (c == '\n')
count = count + 1
printf ("%d", count)

Uso calloc para tamaño y dirección del puntero
ptr = (float *) calloc

Ciclo for para calcular el número máximo
for (i = 0; i < count; i++)

Se guarda el valor del puntero en nom-max
y se verifica si se cumple la condición (recursivo)
nom-max = *ptr
*(ptr + i) = *(ptr + i + 1)
*(ptr + i + 1) = nom-max

Libero memoria free (ptr)

promedio.c // Semra 9 // Guadalupe Florien

Inicio

Declaro variable archivo `*FILE *archivo;`

Declaro variable enteros y de punto flotante `int i, c, cant = 0; float sum = 0; *ptr`

Abro archivo `archivo = fopen (".txt", "r")`

Verifico si archivo existe `if (archivo == NULL) return 0;`

Conteo líneas del archivo contando los caracteres e incrementando conteo si empieza una nueva línea.
`for (c = getc (archivo); c != EOF; c = getc ())
if (c == '\n') cant++;`

Inicio ciclo para leer los números y guardarlos en el apuntador en array

`for (i = 0; i < cant; i++)`

Lee ~~los~~ números `fscanf (txt, ptr + i)`

Sumo valores guardados `sum += *(ptr + i)`

Divido total de valores sumados en `*(ptr + i)` entre el número de líneas del archivo `sum /= cant`

Imprimo promedio `printf`

Libero memoria `free (ptr)`

Cierro archivo `fclose (archivo)`

Fin