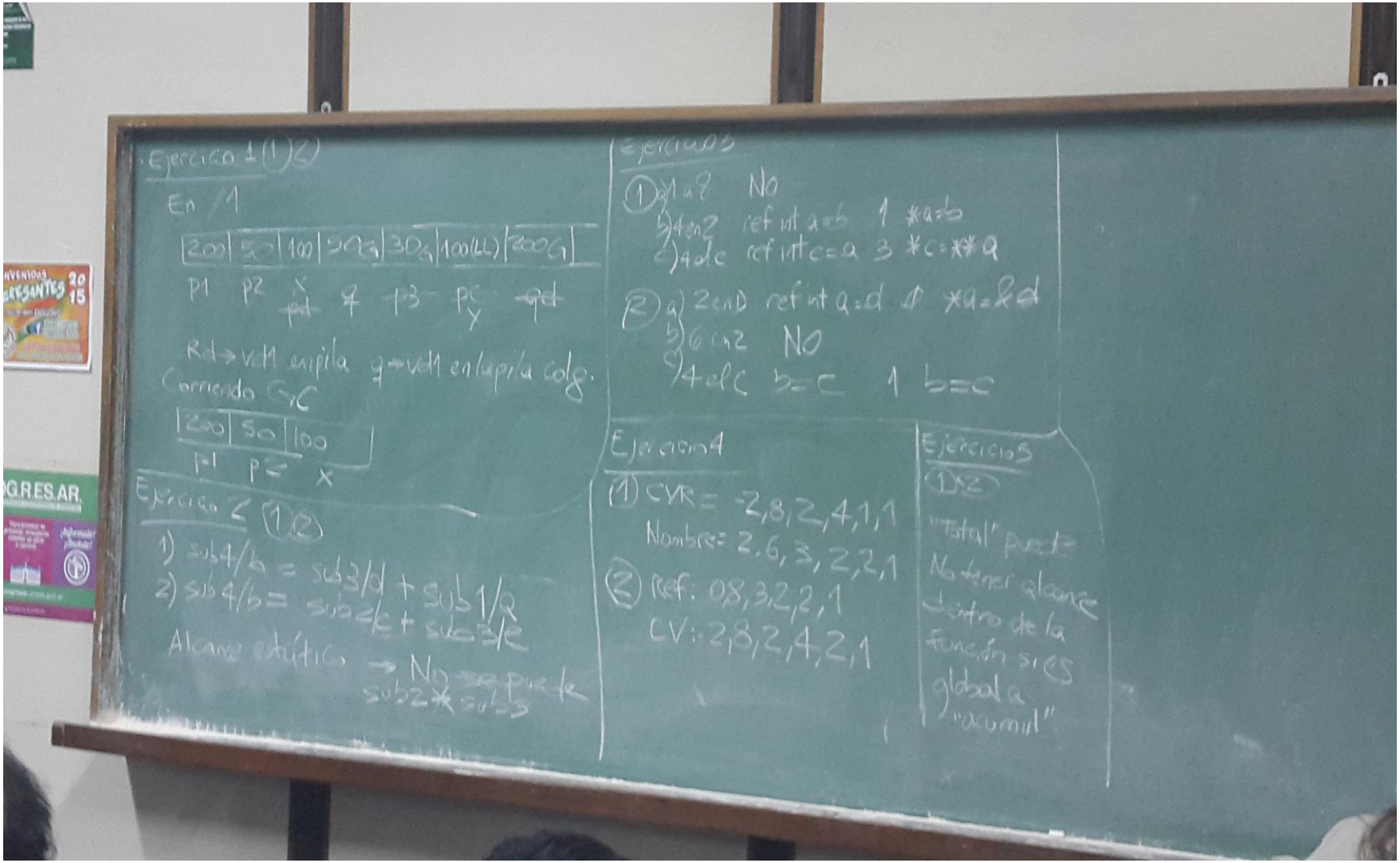
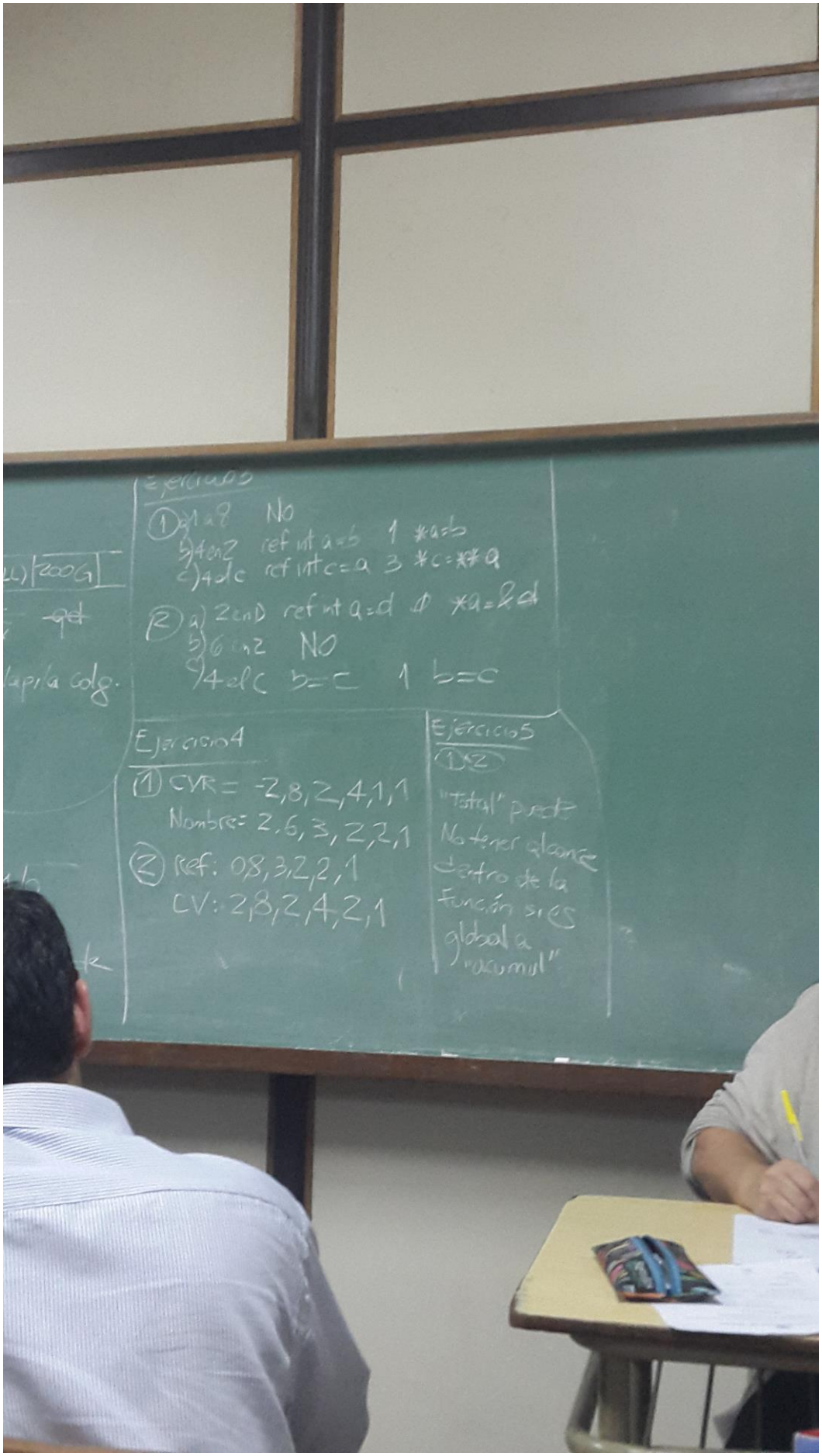

Ejercicio 1 (1) (2)
En /1
200 | 50 | 100 | 503 | 303 | 100(66) | 200(1)
P1 P2 x P3 P5 qd
Rd → vd1 en pila y → vd1 en la pila colg.
Corriendo C1C
200 | 50 | 100
P1 P2 x
Ejercicio 2 (1) (2)
1) sub4/b = sub3/d + sub1/a
2) sub4/b = sub2/c + sub3/e
Alcance estático → No se puede sub2 * sub3
Ejercicio3
(1) a? No
b) 4en2 ref int a=b 1 *a=b
c) 4de ref int c=a 3 *c=**a
(2) a) 2en0 ref int a=d 1 *a=&d
b) 6en2 NO
c) 4de c b=c 1 b=c
Ejercicio4
(1) CVR = -2,8,2,4,1,1
Nombre: 2,6,3,2,2,1
(2) Ref: 08,3,2,2,1
CV: 2,8,2,4,2,1
Ejercicio5
(1) (2)
"total" puede no tener alcance dentro de la función si es global a "acumul".


Ejercicios
1) a) 8 No
b) 4 en 2 ref int a=b 1 *a=b
c) 4 de c ref int c=a 3 *c=**a
2) a) 2 en D ref int a=d 1 *a=&d
b) 6 en 2 NO
c) 4 de c b=c 1 b=c
Ejercicio 4
1) CVR = -2,8,2,4,1,1
Nombre= 2,6,3,2,2,1
2) Ref: 08,3,2,2,1
CV: 2,8,2,4,2,1
Ejercicio 5
1) 2
"Total" puede No tener alcance dentro de la función si es global a "acumul"

Parcial de Lenguajes y Compiladores — Tema 2 — 17-11-2014

Apellido y Nombre: Montero Diego — DNI: 32573721 — Email: nicoemontero@hotmail.com

1/ Luego de la ejecución del procedimiento "Principal" en un determinado lenguaje que utiliza el algoritmo de "marcado y barrido":

```
Principal ( )
  int n
  int *x
  int *q
  int *y
  n=30
  DO( z )
  ...
  ...
  ...

Function Z
 int *punt1
 int *punt2
 int v1[1..n,1..10]
 punt1=malloc(200)
 punt2=malloc(50)
 DO( b )
 -- [ 1 ]
 end

Function B
 n=20
 int v2[1..10,1..n]
 int *punt3
 x=malloc(100)
 q=malloc(50)
 punt3=malloc(30)
 DO( c )
 end

Function C
 int *pc
 int *qc
 int vc1=500
 pc=malloc(100)
 y=pc
 DO( d )
 end

Function D
 int *pd
 int *qd
 int *rd
 int vd1=1000
 pd=x
 rd=&vd1
 qd=malloc(200)
 q=rd
 free y
 end
```

Responder:

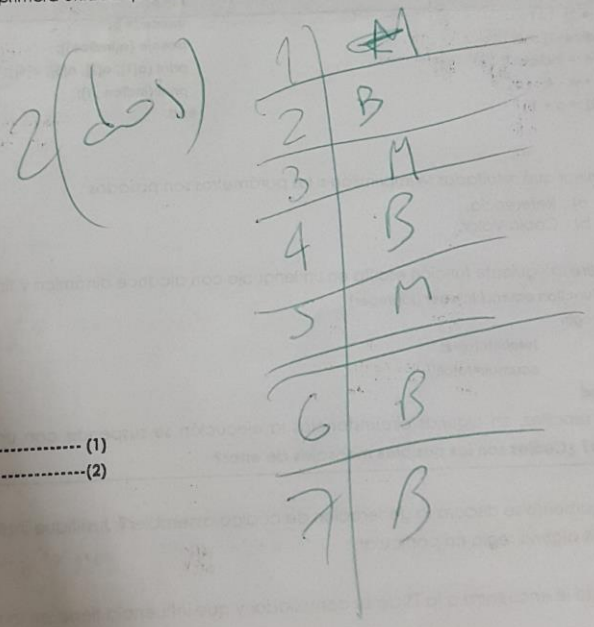
a) Indicar si se genera garbage o punteros colgados en el punto (1) **indicando claramente** las variables involucradas en cada paso. Justificar representando gráficamente el estado del heap

b) En caso de que se haya generado garbage, utilice el GC luego de (1) indicando las direcciones de comienzo de cada variable en el heap y representando gráficamente.
Si no se generó garbaje, proponga una modificación en la función D, que los genere y represente gráficamente.

2/ Siendo la regla de alcance de un determinado programa:
"Los identificadores se buscan en el ambiente local, y en caso de no encontrarse allí, se buscan en el ámbito de la unidad llamadora hasta la primera unidad que fue invocada (main)"

```
procedure main
        var c,d: integer
        procedure sub1
                var a,b: integer;
                {
                 call sub4;
                }
        procedure sub2
                var c,d: integer;
                {
                 call sub3;
                }
        procedure sub4
                var b: integer;
                procedure sub3
                var d,e: integer;
                {
                .....
                b:=d+a; ---------- (1)
                b := c+e; ----------(2)

                }
        {
                call sub2;
        }
{       .....
 call sub1;
}
```

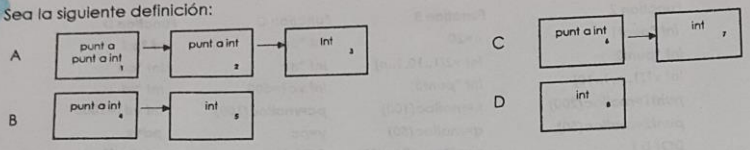


Parcial de Lenguajes y Compiladores — Tema 2 — 17-11-20

Apellido y Nombre: ..... DNI:..... Email:.....

a) Se pide mencionar, ***si es posible***, qué variables son las referenciadas en los puntos indicados en el programa. Justifique. *(ej. nombre_unidad/nombre_variable)*
b) Idem se utilizara la regla de alcance estático

3/ Sea la siguiente definición:



Realizar las siguientes asignaciones *en los casos que sean posibles*, en el lenguaje ALGOL y C, indicando la cantidad de desreferencing para ALGOL.

a) 2 apunte a D
b) 6 apunte a 2
c) 4 apunte al entero que apunta C

4/ Considerando el siguiente fragmento de programa en pseudocódigo, y teniendo en cuenta el lenguaje utiliza conversiones implícitas:

```
var indice, r: integer;
var a: array [1..4] of integer;
var ref ref p: integer;
procedure pasaje( integer z, integer w)
begin
 a[1] := 2;
 a [indice+1] := 8;
 indice := indice+1;
 z := z + w - 4;
 a[w+2] := p + 1 ;
end;
```

```
--MAIN--
begin
 a[1] := 1, a[2] := 2, a[3] := 3, a[4] := 4;
 (ref ref integer) p := 1;
 r := p;
 indice := p;
 pasaje (a[indice]);
 print (a[1], a[2], a[3], a[4]);
 print (indice , r);
end.
```

Determinar qué resultados se imprimirán si los parámetros son pasados

a) Referencia.
b) Copia Valor.

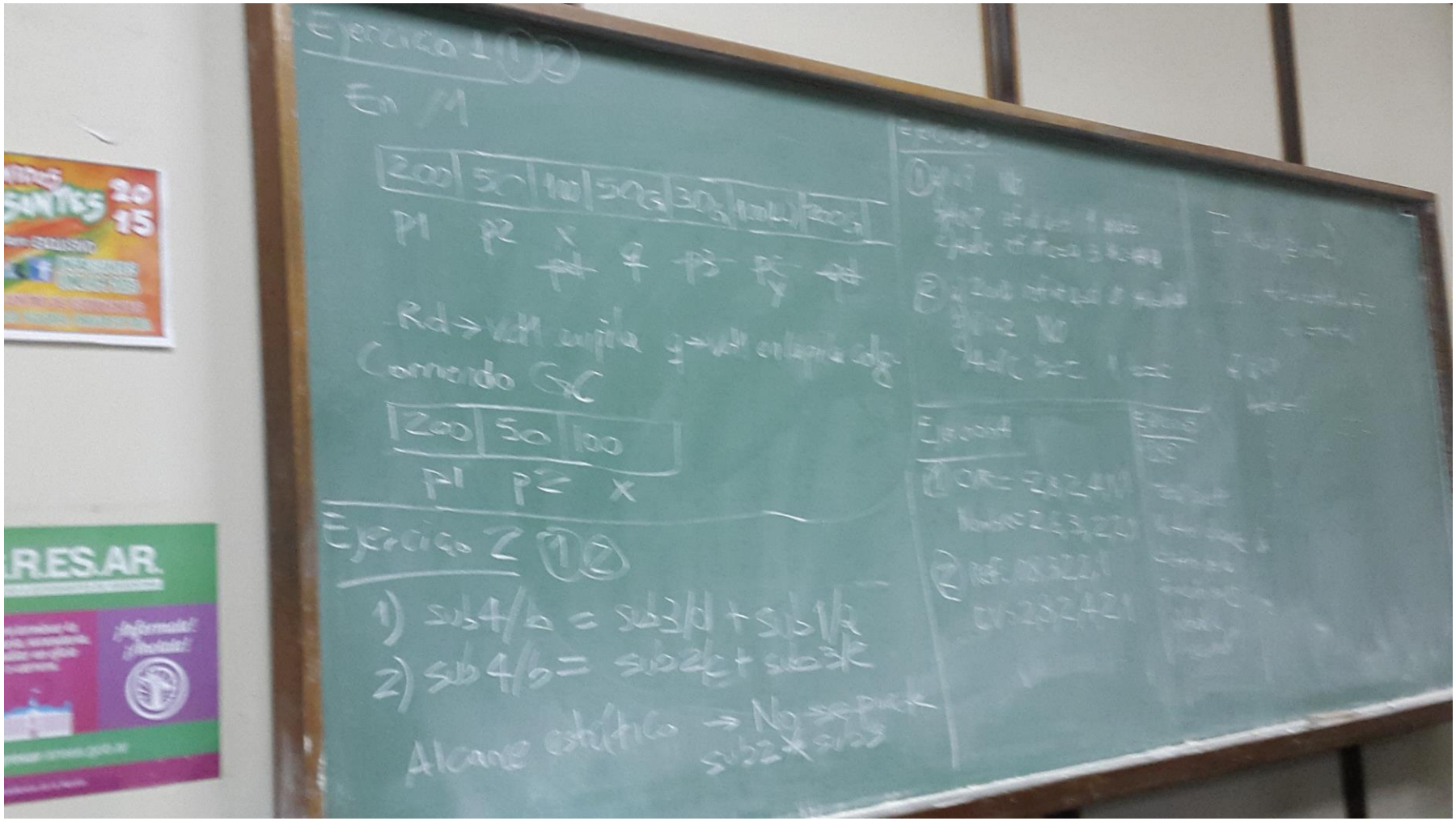
5/ Considere la siguiente función escrita en un lenguaje con alcance dinámico y tipos estáticos:

```
Function acumul:integer (z:integer)
begin
        total:=total+z;
        acumul:=total;
end
```

Pese a su sencillez, en algunas circunstancias la ejecución se suspende con un mensaje de error. **¿Por qué ocurre esto? ¿Cuáles son los posibles mensajes de error?**

6/ ¿En qué momento se dispara la generación de código assembler? Justifique indicando si depende de la ejecución de alguna regla en particular.

7/ ¿Qué utilidad le encuentra a la TS de su compilador y que influencia tiene en la regla E:=E+T?


Ejercicio 1
Ejercicio 2
1) sub4/a = sub3/d + sub1/R
2) sub4/b = sub2c + sub3c