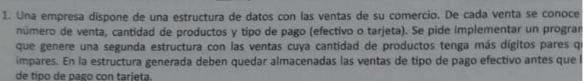
- 4. Calcule el tiempo de ejecución del programa del punto 3. Mostrar los valores intermedios para llegar a resultado y justificar.
- 5. Indique Verdadero o Falso. Justifique en todos los casos:
 - a. Incluir módulos dentro de un programa implica que el programa es más eficiente que otro progra realiza la misma tarea pero sin utilizar módulos.

```
b. El siguiente programa es válido.
     program ejercicio;
       function auxiliar(val:integer): integer;
          val:= val * val;
          auxiliar:= val;
      procedure calculo(c: Integer; var b:integer);
        begin
       b:= b + c DIV 4;
        end;
   var
    a,b:integer;
   begin
     a:= 16;
    b:= 6;
    calculo(auxiliar(a),b);
```

- c. No siempre es posible declarar un tipo subrango donde su tipo base sea cualquiera de los tipos simple en la teoría.
- Un programa que utiliza un repeat until puede reescribirse utilizando un while.
- e. La comunicación entre el programa y los módulos no sólo se puede hacer utilizando parámetros.

Apellido y Nombre ..



 Dados los siguientes programas indique para cada uno si son válidos o no. Además, analice si considera que funcionamiento en ambos programas es el mismo o no. JUSTIFIQUE

```
B
Program uno;
                                     Program dos;
                                      Var
   datos = array [1..100] of
                                       d: array [1..100] of integer;
integer;
 Var
  d: datos:
                                       // Operaciones e invocaciones a
                                     módulos
Begin
                                         para cargar y recorrer el
  //Operaciones e invocaciones a
                                     vector d
módulos
                                      End.
   para cargar y recorrer el
vector d
End
```

 Calcule e indique la cantidad de memoria estática y dinámica que utiliza el siguiente programa. Mostrar lo valores intermedios para llegar al resultado y justificar.

```
program ejercicio3;
                                                     Char
                                                              1 byte
 type info = record
                                                     Integer
                                                              6 bytes
             nombre: string;
                                                     Real
                                                              10 bytes
             nota: integer;
            datos: ^integer;
                                                     Boolean
                                                              1 byte
          end:
                                                     String
                                                              Longitud + 1 byte
      vector = array [1..100] of info;
                                                     Puntero
                                                              4 bytes
 war v: vector; i,j: integer; e: info;
 begin
 read(e.nombre);
 11:=0;
 while (i < 100) and (e.nombre <> 'ZZZ') do
   begin
     read(e.nota);
     e.datos:= nil;
     1:= 1 + 1;
    v[1]:= e;
     read(e.nombre);
 3
 " end;
ofor j:= 1 to i do begin
new(v[j].datos);
n v[j].datos^:= v[j].nota MOD 10;
end:
end.
```