75.01 Computación
Curso de Verano 2013
Primer Parcial 14/2/2013

FACULTAD DE INGENIERIA
Universidad de Buenos Aires

Nombre y Apellido: ..... Padrón: .....

Dirección de email: .....

B

Ejercicio 1) Desarrollar un programa en pascal para leer dos números N y M que deben ser enteros positivos menores a 200 y luego leer dos vectores de elementos enteros **A** y B con N elementos cada uno. Calcular el vector resultante de **A*M+B**, y el producto escalar de **A*B**.

Ejercicio 2) Dados dos números, A= 01111101 escrito en base 2 y B=7B escrito en base 16. Realizar:

B-

d) A-B Utilizando la notación de complemento a dos.
e) Expresar –B en notación de complemento a dos de ocho bits
f) Expresar –B en notación de complemento a uno de ocho bits

(Todos los cálculos deben ser entregados con la hoja del parcial)

B--

Ejercicio 3) Desarrollar un procedimiento en pascal que dada una matriz A de NxN y un numero M que representa el numero de una fila, muestre por pantalla los valores pares de dicha fila ordenados ascendentemente.

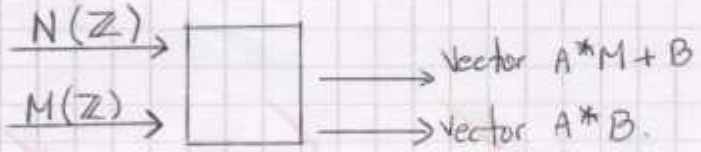
Para aprobar se necesita tener el 60 % de los ejercicios correctamente resueltos.

NOTA: Escriba en letra de imprenta, clara y prolija. Numere las hojas entregadas y repita sus datos en la primera de las hojas.

FECHA 14-2-13

EJERCICIO 1:

Análisis del problema:



N y M enteros > 0. y < 200. También se ingresan los vectores A y B de N elementos c/u.

Debe haber un procedimiento para ingresar N y M, y A y B. (previamente inicializamos A y B)

↳ No necesitás para este caso

```
program ej1;
uses crt; // Declaración de librerías.
const
   MAX = 200; // máx físico de los elementos del vector.
type
   tvector: array[1..MAX] of integer;
var
   A,B,Resultado1: tvector; proc. y funciones
   N,M: integer;
   A,B: char;
procedure obtenerNumeros(var mL: integer; var num: integer);
// mL: máximo lógico de elementos del vector.
var numero, number: integer;
begin
   write('Ingrese el numero de elementos de las vectores A y B
      (>0 y <200):');
   readln(numero); if(numero>0 and <200) then mL:= numero;
   writeln('Ahora ingrese un numero entero (>0 y <200): ');
   readln(number);
   if(number>0 and <200) then num:= number;
end;
```

Mejor luego de la declaración de

Necesitás un ciclo! hasta que el nro sea válido

{Agregué las validaciones}

```
procedure inicializar (var V:tvector ; mL : integer);
Var
   i: integer; // Para recorrer el vector.
begin
   for i:= 1 to mL do
      V[i]:= 0;
end;

procedure cargarVector (var W:tvector ; mL: integer ; vec:char);
Var
   i:integer;
   number: integer;
begin
   writeln('Ingrese los elementos del vector ', Vec, '.');
   for i:= 1 to mL do
   begin
      readln (number);
      W[i]:= number;
   end;
end;

procedure productoAMmasB (var A:tvector ; Var B: tvector; mL:integer; numb:integer;
 var R: tvector);     // R para el resultado. (A*M + B).

Var
   i,j,K:integer
begin
 j:= 1;
 K:= 1;  {inicializamos}
   for i:=1 to mL do
   begin
      R[j]:= A[i]* numb + B[K]
      inc(j);
      inc(K);
   end;
```

NOTA

FECHA 14-2-13

```
function productoescalar(var X:tvector; var Y:tvector; mL:integer):integer;
Var                          // X = vector A.
   i, cont:integer;          // Y = vector B
begin
   cont:=0;  // inicializamos
   for i:=1 to mL do
      cont:= cont + X[i] * Y[i];
   productoescalar:= cont;
end;
```

→ ①

```
beg.n  // Bloque principal.
   obtenerNumeros(N,M);
   inicializar(VA, N);
   inicializar(VB, N);
   cargarVector(VA, N, A);
   cargarVector(VB, N, B);
   productoAMmasB(VA, VB, N, M, Resultado1);
   mostrarproductoAMmasB(Resultado1, N);
   Writeln('El producto escalar A*B es: ', productoescalar(VA,VB,N));
   readkey
end.
```

B--

No es necesario

MB

A y B para que aparezca en pantalla qué vector hay que cargar.

B-

B

① A continuación, van los dos procedimientos que faltan:

```
procedure mostrarproductoAMmasB(var Res:tvector; mL:integer);
Var
   i:integer;
begin
   Writeln('El producto A*M+B es: ');
      for i:=1 to mL do
         writeln(Res[i]);
         writeln('');
end;
```

14 - 2 - 13

EJERCICIO 2:

$A = 01111101]_2$

$B = 7B]_{16}$

d) A - B en complemento a dos.

$A = 01111101]_2$ $(n = 8)$

↓ MSB, al ser cero, el nro es positivo, el valor en módulo del patrón de Bits $(n-1)$ no se modifica. ✓

Para B: primero obtenemos el nº en base 10. Para ello: ✓

$$7B]_{16} = 7.\underbrace{16^1}_{16} + 11.\underbrace{16^0}_{1} = 112 + 11 = 123]_{10}$$

(B = 11)

C. AUX.

$$\begin{array}{r} {}^{4}16 \\ \times\ 7 \\ \hline 112 \end{array}$$

Ahora pasamos $123]_{10}$ a base 2. Para ello:

123 | 2
1/ 61 | 2
1/ 30 | 2
0/ 15 | 2
1/ 7 | 2
1/ 3 | 2
1/ 1 ✓

Entonces: $123]_{10} = 1111011]_2$ como trabajamos con 8bits, tomamos de 1 nro binario recién calculado, y lo pasamos a C2. ✓

1 ↓ _ _ _ _ _ _ _

Por ser negativo

el C2 es el C1 + 1. El C1 es aplicarle un NOT al patrón de bits. Entonces: NOT(1111011) = 0000100 por lo que el número -B representado en complemento a dos es (C. AUX) 10000101.

Procedemos a realizar A - B. Con esta representación, podemos sumar restando.

```
  1 1 1 1 1 1
  0 1 1 1 1 0 1
+ 1 0 0 0 0 1 0 1      } o/p'
-----------------
1 0 0 0 0 0 0 1 0
```

```
  C. AUX
  0 0 0 0 1 0 0
+         1
---------------
  0 0 0 0 1 0 1
```

e) -B expresado en C2 ya fue realizado en (d).

-B en C2 es 10000101

↓

MSB denota signo negativo

f) -B expresado en C1 también fue realizado en (d).

-B en C1 es 10000100.

↓

MSB denota signo negativo

14-2-13

EJERCICIO 3:

procedimiento para: análisis del problema:

→ M fila de la matriz A (los nº pares) y ordenarlos ascendentemente.

Como hay que ordenar, hacemos un procedimiento aparte para hacer el intercambio de valores. Utilizamos ordenamiento por burbujeo.

```
procedure intercambio (var A,B: integer);
var
   aux: integer
begin
   aux:= A
   A:= B
   B:= aux
end;
```

{Entonces el proc. que ordena invocará al intercambio}.

{la Matriz A es cuadrada por ser NxN}

```
procedure busca pares (var A: tmatriz; M: integer; var R: tvector);
var
   i,j: integer;
begin
   i:= 1;
   for j:= 1 to N do
      if (A[M,j] mod 2)=0 then
      begin
         A[i]:= A[M,j];
         inc(i);
      end;
end;
```

MLi

{A[i,j] i: para fila, j: para columna}

{N se supone declarada en la parte de constantes}

No es necesario inicializarlo

{suponiendo que tenemos al vector R inicializado}

{Este procedimiento almacena los nros pares en el vector R, de long. física máxima N}

tenemos que devolver i-1, pues es la cantidad de elementos pares que cargamos en el vector R

```
procedure ordena Ascendente (var R: tvector);
```

MLi

MLi := i-1

```
var
  i,j:integer;
begin
    for i:= 1 to N-1 do
        for j:=1 to N-i do
            if (R[j] > R[j+1]) then
                intercambio (R[j], R[j+1]);
end;
```

Ojo! Acordate, elementos adyacentes! porque elegiste burbujeo

Ahora mostramos los datos en pantalla:

```
procedure mostrar ordenado (var R: tvector ; M:integer);
 var
    i: integer
begin
    writeln('Los ---> valores pares de la tira', M,' ordenados
      ascendentemente son: ');
      for i:= 1 to N do
          writeln (R[i]);
end;
```

MLi

De esta forma, un programa principal puede invocar a los procedimientos y pasarle info. mediante los parámetros.