

PARCIAL para PROMOCION PROGRAMACION I (27/11/2024)

ALUMNO/A: Acevedo Frías, Valentina..... CALIFICACION: 10.....

Para aprobar este examen deben obtenerse al menos 7 puntos.

Debe tener al menos el 50% de cada ejercicio

De aprobarse, la nota final surgirá del promedio de la calificación de este

examen con el promedio de los parciales.  $NF = (P1 + P2 + P3) / 3 \geq 7$ .

→ 9 Promociones

**Ej1-(7 puntos)**

Desarrollar un programa en lenguaje Pascal, CLARO y EFICIENTE, que utilizando archivos, metodología de Corte de Control, Apareo y/o lo que considere necesario, resuelva el siguiente problema:

Una imprenta cuenta con un archivo TRABAJOS.DAT donde se han almacenado los trabajos realizados, cada registro contiene:

- Fecha (alfanumérico de 8 aaaammdd, 1er. campo de secuencia, clave secundaria)
- Tipo de impreso (1..50, 2do. campo de secuencia, clave secundaria)
- Color (s/n)
- Tamaño (c-chico, m-mediano, g-grande)

Además, posee un archivo de texto con la descripción del Tipo de impreso (código de 1 a 50, almacenado en ese orden).

El costo se calcula sobre una base de \$1.500. Si está entre los tipos 20 y 35 se le adiciona un 10%, desde 36 en adelante un 25%. Si el tamaño es mediano un 5% y si es grande un 15% extra.

Se pide emitir el siguiente listado:

Fecha: aaaammdd	T.Chico	T.Mediano	T.Grande	Costo
Tipo de impreso	9999	9999	9999	\$ 99,999.99
XXXXXXXXXXXXXXXX	9999	9999	9999	\$ 99,999.99
XXXXXXXXXXXXXXXX				

-----  
Porcentaje Color sobre el total de impresos de la fecha: 99.99%

Fecha: aaaammdd	T.Chico	T.Mediano	T.Grande	Costo
Tipo de impreso	9999	9999	9999	\$ 99,999.99
XXXXXXXXXXXXXXXX	9999	9999	9999	\$ 99,999.99
XXXXXXXXXXXXXXXX				

-----  
Porcentaje Color sobre el total de impresos de la fecha: 99.99%

-----  
Fecha con mayor cantidad de impresos: aaaammdd con 999 impresos

**3 Ej 2-(3 puntos)**

Este programa lee una matriz MAT de NxM enteros y un vector PRIM de P números primos, calcula cuántas filas tienen todos sus números múltiplos de algún elemento del vector.

Program parcial3;

```
Type
  TMAT= array [1..20, 1..20] of integer;
  TVEC= array [1..10] of integer;
```



```

Var
  Mat:TMAT;
  PRIM:TVEC;
  i,j,n,m,P,k: byte;
  Cont,ContP,CantFilas,Mul:integer;

Begin
  Writeln('Ingrese n y m'); Readln(n,m);
  For i:=1 to n do
    For j:= 1 to m do
      Readln(Mat[i,j]);
  Writeln('Ingrese P'); Readln(P);
  For k:=1 to P do
    Readln(Prim[k]);
  CantFilas:=0;
  ContP:=0;
  For i:=1 to n do
    begin
      ContP:=0;
      For j:= 1 to m do
        Begin
          Mul:=0;
          For k:=1 to P do
            if Mat[i,j] mod PRIM[k]=0 then
              Mul:=Mul+1;

          If Mul>0 then
            ContP:=ContP+1;
          end;
        if ContP=M then
          CantFilas:=CantFilas+1;
        ContP:=0;
      end;
    end;
  Writeln('Hay ',CantFilas,' filas con todos sus elem multiples de algún elem del
vector');
  READLN;
End.

```

Se pide: reescribirlo modularizando y haciendo los cambios necesarios para mejorar su eficiencia.



an parcial promocion;

(1) 15

ST

max\_tipo = 50; Precio\_base = 1500;

obs: usar mayusculas

type

STB = string[8];

ST30 = string[30];

regT = record

Fecha: STB;

Tipo-impreso: byte;

color, Tamano: Char;

end;

regT = record

Tipo-impreso: byte;

regT = ST30;

AT = File of regT;

TV = array [1..50] of ST30;



```

Procedure gen-vec (var VTipo: TV);
var
  arch: text;
  i: byte;
begin
  assign(arch, 'Tipo-Impreso.TXT');
  reset(arch);
  i := 0;
  while not eof(arch) do
    begin
      i := i + 1;
      readln(arch, VTipo[i]);
    end;
  close(arch);
end;

```

```

Function costo (tipo-impreso: byte; tamaño: char): real;
var
  aux-costo: real;
begin
  aux-costo := precio-base;
  if tipo-impreso in [20..35] then
    aux-costo := aux-costo + 0.1 * precio-base
  else
    if tipo-impreso >= 36 then
      aux-costo := aux-costo + 0.25 * precio-base;
    if tamaño = 'M' then
      aux-costo := aux-costo + 0.05 * precio-base;
    else
      if tamaño = 'G' then
        aux-costo := aux-costo + 0.15 * precio-base;
      costo := aux-costo;
    end;
  end;

```



procedure CORTE (VAR trabajos : AT, VTIPOS : TV);

VAR

RT : reg T;

muyor-contimp, cantimp, cantcolor, cant-c, cant-m, cant-g, tipo-act : byte;

acum-costo : real;

fecha-muyorimp; fecha-act : str;

begin

reset (trabajos);

muyor-cantimp := 0;

read (trabajos, RT);

while not eof (trabajos) do

begin

cantcolor := 0;

cantimp := 0;

fecha-act := RT.fecha;

writeln ('Fecha:', fecha-act);

writeln ('Tipo de Impreso T.chico T.mediano T.grande costo');

while fecha-act = RT.fecha do

begin

cant-c := 0;

cant-m := 0;

cant-g := 0;

acum-costo := 0;

tipo-act := RT.tipo-impreso;

while (fecha-act = RT.fecha) and (tipo-act = RT.tipo-impreso) do

begin

case of RT.tamano do

'c': cant-c := cant-c + 1;

'm': cant-m := cant-m + 1;

'g': cant-g := cant-g + 1;

end;

acum-costo := acum-costo + costo (RT.tipo-impreso, RT.tamano);

if RT.color = '3' then

cantcolor := cantcolor + 1;

read (trabajos, RT);

end;



```

(9) writeIn (vtipos [tipo-act], cant-c, cant-m, cant-g, acum-costo),
    cantimp := cantimp + cant-c + cant-m + cant-g,
end,
writeIn ('Porcentaje color sobre el total de impresos de la fecha:',
    cantcolor * 100 / cantimp),
if cantimp > mayor-cantimp then
    begin
        mayor-cantimp := cantimp;
        fecha-mayorimp := fecha-act.
    end,
end,
writeIn ('Fecha con mayor cantidad de impresos:', fecha-mayorimp, ' con',
    mayor-cantimp, ' impresos'),
close (trabajos),
end,

```



```
var  
  trabajos: AT;
```

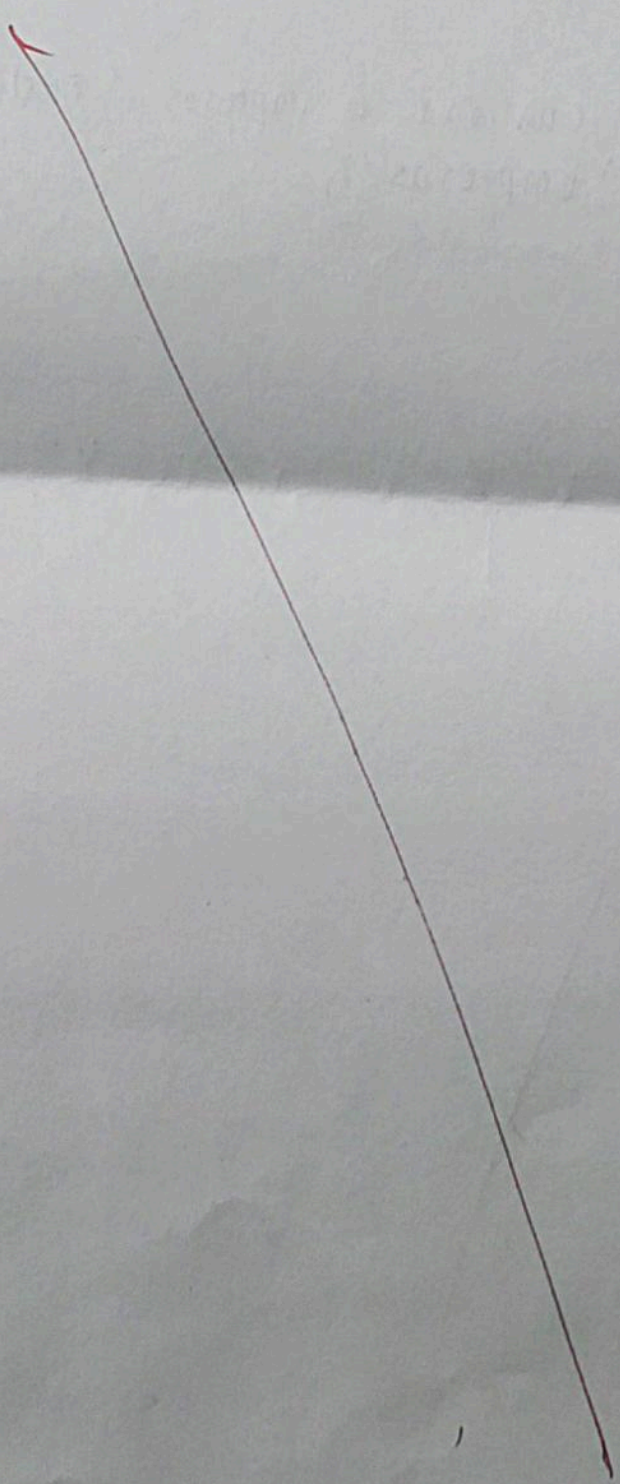
```
  vtipos: TV;  
begin
```

```
  assign(trabajos, 'trabajos.dat');
```

```
  gen-vec(vtipos);
```

```
  corte(trabajos, vtipos);
```

```
end.
```





an ejercicio 2;

type

Obs: usar constantes

(4) 15

THAT = array [1..20, 1..20] of integer;

TVEC = array [1..10] of integer;

Procedure amor-mat (var mat: THAT; var n, m: byte);  
var  
i, j: byte;

begin  
writeh('Ingrese n y m');  
readh(n, m);  
for i := 1 to n do  
for j := 1 to m do  
readh(mat[i, j]);  
end;

Procedure armas-prim (var PRIM: TVEC; var P: byte);  
var  
k: byte;

begin  
writeh('Ingrese P');  
readh(P);  
for k := 1 to P do  
readh(PRIM[k]);  
end;



```
function elen_mult(mat: TMat; PRIM: tVec; i, j, p: byte): boolean;
```

```
var  
  k: byte;
```

```
begin
```

```
  k := 1;
```

```
  while (k < p) and (mat[i, j] mod PRIM[k] <> 0) do
```

```
    k := k + 1;
```

```
  elen_mult := mat[i, j] mod PRIM[k] = 0; dos: evaluating 2 vecs  
  guards k < p
```

```
end;
```

```
function fila_mult(mat: TMat; PRIM: tVec; i, m, p: byte): boolean;
```

```
var  
  j: byte;
```

```
begin
```

```
  j := 1;
```

```
  while (j < m) and not elen_mult(mat, PRIM, i, j, p) do
```

```
    j := j + 1;
```

```
  fila_mult := elen_mult(mat, PRIM, i, j, p);
```

```
end;
```

```
function Cuantos_FilasMult(mat: TMat; PRIM: tVec; n, m, p: byte): byte;
```

```
var  
  i, aux: byte;
```

```
begin
```

```
  aux := 0;
```

```
  for i := 1 to n do
```

```
    if fila_mult(mat, PRIM, i, m, p) then
```

```
      aux := aux + 1;
```

```
  cuantos_FilasMult := aux;
```

```
end;
```



AT: TMat;  
PM: TVEC;  
n,m,p: byte;

begin

armar-mat (mat, n, m),

armar-prim (pm, p),

writeln('hay', cuantos-filas-muit (mat, prim, n, m, p), 'Filas con todos sus  
elem multiples de algun elem del vector'),

end.

