```
Program parcial;
Const
  dF = 3; ValorAlto = 9999;
type
  datoMae = record
    codigo: integer;
    nombre: string;
    descripcion: string;
    codigoBarras: string;
    categoria: integer;
    stockActual: integer;
    stockMin: integer;
  end;
  datoDet = record
    codigo: integer;
    cant: integer;
    descripcion: string;
  maestro = file of datoMae;
  detalle = file of datoDet;
  vectDetalle = array[1..dF] of detalle
  VectRegistro = array[1..dF] of datoDet;
var
  detalles: VectDetalles;
  mae: maestro;
begin
  Asignar(detalles, mae);
  Resolucion(detalles, mae);
end.
```

```
Procedure Asignar(Var detalles: VectRegistros; Var mae:maestro);
Var
  i:integer; istring: string;
begin
    for i:=1 to df do begin
      Str(i,string);
      Assign(detalles[i],'detalle'+istring+'.dat');
    end;
    Assing(maestro,'maestro.dat');
end;
Procedure Resolucion(Var detalles: VectDetalles; var mae:maeyd;
Var
  registros: VectRegistro; minimo: datoDet; actual: datoMae;
  resumen: text; Informen: text;
begin
  Assign(resumen,'resumen.txt');
  AbrirYLeer(detalles, registros);(resumen,mae,informe);
  Min(detalles, registros, minimo);
  While(minimo.codigo <> Valor Alto) do begin
    read(mae, actual);
    while (actual.codigo <> minimo.codigo) do read(mae,actual);
    while(actual.codigo=minimo.codigo) do begin
      actual.stockActual:= actual.stockActual-minimo.cant;
    (Min(detalles,registros, minimo);
      if(actual.stockActual <0)then begin
        if(actual.stockActual*-1 >min.cant)then
          writeln(resumen, min.cant),
        else writeln(resumen, min.cant-abs(actual.
          stockActual));
      end;
```

```
        end;
        Seek(mae, FilePos(mae)-1);
        Write(mae, actual);
    end; → if(actual.stockMin < actual.stockActual) then writeln(informe, actual.
cerrar(detalles, registros, resumen); codigo, ' ', actual.categoria);
end;
Procedure leer(var det: detalle; var reg: datDet);
begin
    if(not eof(det)) then read(det, reg)
    else reg.codigo := valorAlto;
end;
Procedure Min (var detalles: vecDetalle; var regs: vecRegs
tro; var minimo: datDet;
var i: integer; pos: integer;
begin
    min.codigo := valorAlto;
    for i := 1 to dF do
        if(regs[i].codigo < min.codigo) then begin
            min := regs[i];
            pos := i;
    if(min.codigo <> valorAlto) then leer(detalles[pos], reg[pos]
end;
```

```
procedure AbrirYLeer(var dets:vecDetalle; var regs:vecRegis
tros; var resumen:text; var mae:maestro; var i:integer;
var informe:text);
begin
  Reset(mae);
  for i:=1 to 8 do begin
    Reset(dets[i]);
    leer(dets[i], regs);
  end;
  rewrite(resumen); rewrite(informe);
end;
procedure Leer(var dets:vecDetalle; var mae:maestro; var re
sumen:text; var informe:text);
var i: integer;
begin
  for i:= 1 to 8 do close(dets[i]);
  close(mae);
  close(resumen); close(informe);
end;
```

2

2[0(238)3(547)1(729)4]

0[(145)] → 3[(238),(402)(512)] → 1[(679)] → 4[(729)] -1

+500

238, 402, 500, 512 — 238, 500, 547, 729

7[2(547)6]

2[0(238)3(500)5] 6[1(729)4]

0[(145)] → 3[(238),(402)] → 5[(500)(512)] → 1[(679)] → 4[(729)]

|2|3|e3|e5|e2|e6|e7

-145

238, 402

|7|2|0|e0|e3|e2

7[2(547)6]

2[0(402)3(500)5] 6[1(729)4]

0[(238)] → 3[(402)] → 5[(500)(512)] → 1[(679)] → 4

-402

7[2(547)6]

2[0(402)3(512)5] 6[1(729)4]

0[(238)] → 3[(500)] → 5[(512)] → 1[(679)] → 4

|7|2|3|5|e1|e5|e2

3)

+13 (Pos 2)

| | | | | | |
|---|---|---|---|---|---|
| 0 | 66 | – | 0 | 66 | – |
| 1 | 56 | 34 | 1 | 56 | 34 |
| 2 | 46 | – | 2 | 46 | 13 |
| 3 | – | – | 3 | – | – |
| 4 | 81 | – → | 4 | 81 | – |
| 5 | 16 | – | 5 | 16 | – |
| 6 | 61 | – | 6 | 61 | – |
| 7 | 95 | – | 7 | 95 | – |
| 8 | – | – | 8 | – | – |
| 9 | 31 | – | 9 | 37 | – |
| 10 | 32 | 21 | 10 | 32 | 21 |

12, e2

+23 (Pos 1)

| | | | | | |
|---|---|---|---|---|---|
| 0 | 66 | – | | | |
| 1 | 56 | 34 | | | |
| 2 | 46 | 13 | | | |
| 3 | 23 | – → | | | |
| 4 | 81 | – | | | |
| 5 | 16 | – | | | |
| 6 | 61 | – | | | |
| 7 | 95 | – | | | |
| 8 | – | – | | | |
| 9 | 31 | – | | | |
| 10 | 32 | 21 | | | |

11, 12, 13, e3

+42 (Pos 9)

| | | | | | |
|---|---|---|---|---|---|
| 0 | 66 | – | | | |
| 1 | 56 | 34 | | | |
| 2 | 46 | 13 | | | |
| 3 | 23 | – → | | | |
| 4 | 81 | – | | | |
| 5 | 16 | – | | | |
| 6 | 61 | – | | | |
| 7 | 95 | – | | | |
| 8 | – | – | | | |
| 9 | 31 | 42 | | | |
| 10 | 32 | 21 | | | |

19, e9

+75 (Pos 9)

| | | | | | |
|---|---|---|---|---|---|
| → | 0 | 66 | 75 | | |
| | 1 | 56 | 34 | | |
| | 2 | 46 | 13 | | |
| | 3 | 23 | – → | | |
| | 4 | 81 | – | | |
| | 5 | 16 | – | | |
| | 6 | 61 | – | | |
| | 7 | 95 | – | | |
| | 8 | – | – | | |
| | 9 | 31 | 42 | | |
| | 10 | 32 | 21 | | |

19, 110, 10, e0

→ 31 (Pos 9)

| | | | | | |
|---|---|---|---|---|---|
| 0 | 66 | 75 | | | |
| 1 | 56 | 34 | | | |
| 2 | 46 | 13 | | | |
| 3 | 23 | – → | | | |
| 4 | 81 | – | | | |
| 5 | 16 | – | | | |
| 6 | 61 | – | | | |
| 7 | 95 | – | | | |
| 8 | – | – | | | |
| 9 | # | 42 | | | |
| 10 | 32 | 21 | | | |

19, 110, e9