Universidad Nacional de Jujuy

Facultad de Ingeniería

Metodología De La Programación

Trabajo Práctico N° 4

Comisión N°5

Grupo N°5

Gutiérrez Karen

Paredes Mariana

Rivera Lucas

Ejercicio propuesto por el profesor

3. Considerando una lista de productos con los campos código, nombre, precio y tipo (A, B, C), realizar un programa que:

a. Cargue los productos hasta que el código ingresado sea cero.

b. Muestre la lista de productos de un tipo dado.

c. Muestre la lista de productos ordenados por nombre de menor a mayor utilizando el método selección.

d. Muestre los primeros **K** productos de mayor precio ordenados por inserción.

e. Elimine un producto

f. Modifique los datos de un producto **program** TP04p1;

**uses** CRT;

**const**

  MAX=100;

**type**

TProducto = **record**

  Codigo:**integer**;

  Nombre:**string**[30];

  Precio:**real**;

  Tipo:**char**;

**end**;

AProductos = **Array**[1..MAX] **of** TProducto;

**function** BuscarProductoPorCodigo(Productos:AProductos;

  N:**integer**;Buscado:**integer**):**integer**;

**var**

  i:**integer**;

**begin**

  BuscarProductoPorCodigo:=-1;

  i:=1;

**while** (BuscarProductoPorCodigo = -1) and (i<=N) **do**

**begin**

**if** (Productos[i].Codigo = Buscado) **then**

**begin**

        BuscarProductoPorCodigo:=i;

**end**;

     i:=i+1;

**end**;

**end**;

*//Controla que el tipo ingresado sea correcto*

**function** CargarTipo():**char**;

**begin**

**repeat**

    ReadLn(CargarTipo);

    CargarTipo := upCase(CargarTipo);

**if** **not**((CargarTipo='A')**or**(CargarTipo='B')**or**(CargarTipo='C')) **then**

**begin**

       writeln('Tipo incorrecto, debe ser A, B o C, ingrese nuevamente');

       write('->');

**end**;

**until** (CargarTipo='A')**or**(CargarTipo='B')**or**(CargarTipo='C');

**end**;

**procedure** CargarProductos(**var** Productos:AProductos;**var** N:**integer**);

**var**

  i:**integer**;

  Cod:**integer**;

**begin**

     i:=n+1;

**repeat**

        Write('Ingrese el codigo del producto (0 para terminar) ->') ;

        readln(Cod);

**while** (BuscarProductoPorCodigo(Productos,N,Cod)<>-1) **do**

**begin**

          writeln('El codigo ingresado ya existe, por favor ingrese otro');

          write('->');

          readln(Cod);

**end**;

**if** (Cod<>0) **then**

**begin**

**with** productos[i] **do**

**begin**

             Codigo:=Cod;

             write('Ingrese el nombre del producto ->');

             readln(Nombre);

             write('Ingrese el precio del producto ->');

             readln(Precio);

             write('Ingrese el tipo del producto ->');

             Tipo := CargarTipo();

             writeln;

             N:=N+1;

             i:=i+1;

**end**;

**end**;

**until** (Cod=0);

**end**;

**procedure** CrearCabecera();

**begin**

  ClrScr();

  Write('Codigo'); gotoxy(10,1);

  Write('Nombre'); gotoxy(30,1);

  Write('Precio'); gotoxy(40,1);

  Writeln('Tipo');

**end**;

*//Muestra un producto en formato tabla en la fila indicada por y*

**procedure** MostrarProductoTabla(Producto:Tproducto;y:**integer**);

**begin**

**with** Producto **do**

**begin**

    Write(Codigo); gotoxy(10,y);

    Write(Nombre); gotoxy(30,y);

    Write(Precio:5:2); gotoxy(40,y);

    Writeln(Tipo);

**end**;

**end**;

**procedure** MostrarProductosPorTipo(Productos:AProductos;N:**integer**;TipoBuscado:**char**);

**var**

  i,PosicionEnTabla:**integer**;

**begin**

   PosicionEnTabla:=2;

   CrearCabecera();

**for** i:=1 **to** N **do**

**with** Productos[i] **do**

**if** (Tipo=TipoBuscado) **then**

**begin**

         MostrarProductoTabla(Productos[i],PosicionEnTabla);

         Inc(PosicionEnTabla);

**end**;

**end**;

**procedure** Intercambiar(**var** X:TProducto;**var** Y:TProducto);

**var**

  Temp:TProducto;

**begin**

  Temp:=X;

  X:=Y;

  Y:=Temp;

**end**;

*//metodo de seleccion*

**procedure** OrdenarProductosPorNombre(**var** Productos:AProductos;N:**integer**);

**Var**

 i, j,Minimo: **Integer**;

**Begin**

**For** i := 1 **To** N - 1 **Do**

**Begin**

     Minimo:=i;

**For** J := (I + 1) **To** N **Do**

**If** (Productos[j].Nombre < Productos[Minimo].Nombre) **Then**

           Minimo:=j;

       Intercambiar(Productos[i],Productos[Minimo]);

**End**

**End**;

**procedure** MostrarProductosPorNombre (**var** Productos:AProductos;N:**integer**);

**var** i:**integer**;

**begin**

  OrdenarProductosPorNombre(Productos,N);

  CrearCabecera();

**for** i:=1 **to** N **do**

       MostrarProductoTabla(Productos[i],i+1);

**end**;

*//metodo de insercion*

**Procedure** OrdenarProductosPorMayorPrecio(**Var** ListaProductos : AProductos; N : **Integer**);

**Var**

   Top,

   InsersionPos:**integer**;

   Cache: TProducto;

   Found: **Boolean**;

**Begin**

**For** Top := 2 **To** N **Do**

**Begin**

     Cache := ListaProductos[Top];

     InsersionPos := Top - 1;

     Found := **false**;

**While** (InsersionPos >= 1)

      And (**Not** Found) **Do**

**If** (Cache.Precio > ListaProductos[InsersionPos].Precio) **Then**

**Begin**

         ListaProductos[InsersionPos + 1] := ListaProductos[InsersionPos];

         InsersionPos := InsersionPos - 1

**End**

**else**

         Found := **true**;

       ListaProductos[InsersionPos + 1] := Cache;

**End**

**End**;

**procedure** MostrarKProductosPorMayorPrecio (**var** Productos:AProductos;

  N:**integer**;K:**integer**);

**var**

  i:**integer**;

**begin**

    OrdenarProductosPorMayorPrecio(Productos,N);

    CrearCabecera();

**if** (k>n) **then**

       k:=n;

**for** i:=1 **to** K **do**

      MostrarProductoTabla(Productos[i],i+1);

**end**;

**procedure** EliminarProducto(**var** Productos:AProductos;**var** N:**integer**;

  Buscado:**integer**);

**var**

  i,Pos:**integer**;

**begin**

   Pos:=BuscarProductoPorCodigo(Productos,N,Buscado);

**if** (Pos<>-1) **then**

**begin**

**for** i:=Pos **to** N-1 **do**

       Productos[i]:=Productos[i+1];

     dec(N);

     writeln('Producto eliminado correctamente');

**end**

**else**

     writeln('El producto no existe');

**end**;

**procedure** ModificarProducto(**var** Productos:AProductos;**var** N:**integer**;

  Buscado:**integer**);

**var**

  Pos:**integer**;

**begin**

   Pos:=BuscarProductoPorCodigo(Productos,N,Buscado);

**if** (Pos<>-1) **then**

**begin**

**with** Productos[Pos] **do**

**begin**

       writeln('Modifique el nombre (anterior:',Nombre,')');

       write('->');

       readln(Nombre);

       writeln('Modifique el precio (anterior:',Precio:4:2,')');

       write('->');

       readln(Precio);

       writeln('Modifique el tipo (anterior:',Tipo,')');

       write('->');

       Tipo := CargarTipo();

**end**;

**end**

**else**

     writeln('El producto no existe');

**end**;

**function** MostrarMenu():**integer**;

**begin**

  WriteLn('MENU');

  WriteLn('1- Cargar productos');

  WriteLn('2- Mostrar productos por tipo');

  WriteLn('3- Mostrar productos ordenados por nombre');

  WriteLn('4- Mostrar productos con mayor precio');

  WriteLn('5- Eliminar un producto');

  WriteLn('6- Modificar un producto');

  WriteLn('0- Salir');

  Write('->');

  Readln(MostrarMenu);

**end**;

**var**

  ListaProductos:AProductos;

  N,Opcion,Cantidad,Codigo:**Integer**;

**begin**

  N:=0;

**repeat**

     ClrScr;

     Opcion:=MostrarMenu();

**case** Opcion **of**

       1:CargarProductos(ListaProductos,N);

       2:

**begin**

         write('Ingrese el tipo por el que desea filtrar(A,B,C) ->');

         MostrarProductosPorTipo(ListaProductos,N,CargarTipo());

**end**;

       3:MostrarProductosPorNombre(ListaProductos,N);

       4:

**begin**

          write('Cuantos productos desea visualizar? ->');

          readln(Cantidad);

          MostrarKProductosPorMayorPrecio(ListaProductos,N,Cantidad);

**end**;

       5:

**begin**

          write('Ingrese el codigo del producto a eliminar ->');

          ReadLn(Codigo);

          EliminarProducto(ListaProductos,N,Codigo);

**end**;

       6:

**begin**

          write('Ingrese el codigo del producto a modificar ->');

          ReadLn(Codigo);

          ModificarProducto(ListaProductos,N,Codigo);

**end**;

       0:writeln('Fin del programa');

**else**

         writeln('Opcion incorrecta');

**end**;

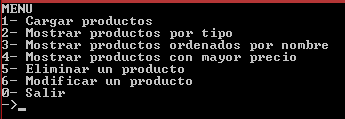
     WriteLn;

     write('Presione una tecla para continuar...');

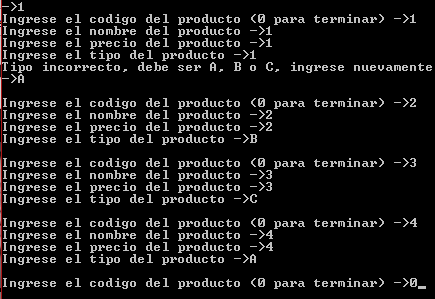
     ReadKey;

**until** Opcion=0;

**end**.



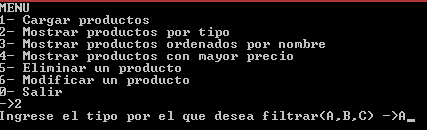
CARGA

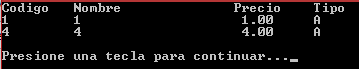


COMPROBACION DE CODIGOS YA EXISTENTES

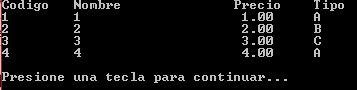


FILTRO POR TIPO

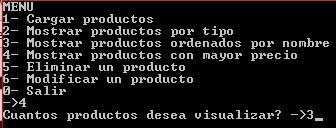


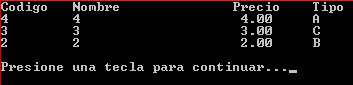


MOSTRAR ORDENADO POR NOMBRE

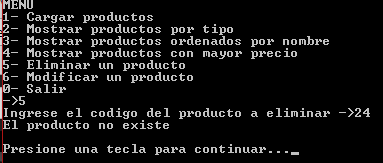


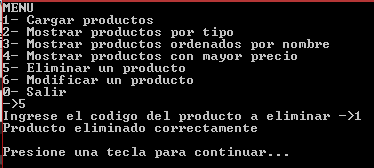
MOSTAR N PRODUCTOS POR MAYOR PRECIO

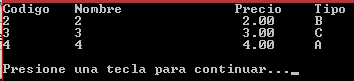




ELIMINAR UN PRODUCTO







MODIFICAR UN PRODUCTO

