```
/*
 * File:   main.cpp
 * Author: Silvia Vargas Cáceres
 *
 * Created on 08 de octubre de 2024, 02:54 PM
 */

#include <iostream>
#include <fstream>
#include <iomanip>

using namespace std;

#define TAM_PRODUCTOS 100
#define TAM_STOCKPROD 300

#include "funciones.h"

int main(int argc, char** argv) {
    int arrStockProd[TAM_STOCKPROD],arrStockAlm[TAM_STOCKPROD],cantStockProdAlm,
            ddIni,mmIni,aaIni,ddFin,mmFin,aaFin,fechaIni,fechaFin;
    double arrProdAlmStockIni[TAM_STOCKPROD],arrProdAlmIng[TAM_STOCKPROD]{},
            arrProdAlmSal[TAM_STOCKPROD]{},arrProdAlmEnv[TAM_STOCKPROD]{},
            arrProdAlmRec[TAM_STOCKPROD]{},arrStockAlmFin[TAM_STOCKPROD];

    cargarArrStockProductos(arrStockProd,arrStockAlm,arrProdAlmStockIni,cantStockProdAlm)
    ;

    mostrarArrStockProductos(arrStockProd,arrStockAlm,arrProdAlmStockIni,cantStockProdAlm
    );
    /*ingresarFechas(ddIni,mmIni,aaIni,ddFin,mmFin,aaFin);
    fechaIni=agruparFecha(ddIni,mmIni,aaIni);
    fechaFin=agruparFecha(ddFin,mmFin,aaFin);*/
    fechaIni=20220101;
    fechaFin=20233112;
    procesarTransaccionesAlmProd(fechaIni,fechaFin,arrStockProd,arrStockAlm,
            cantStockProdAlm,arrProdAlmIng,arrProdAlmSal,arrProdAlmEnv,arrProdAlmRec);
    calcularStockFinalAlmProd(cantStockProdAlm,arrProdAlmStockIni,arrProdAlmIng,
            arrProdAlmSal,arrProdAlmEnv,arrProdAlmRec,arrStockAlmFin);
    ordenarProductos(arrStockProd,arrStockAlm,cantStockProdAlm,arrProdAlmStockIni,
        arrProdAlmIng,arrProdAlmSal,arrProdAlmEnv,arrProdAlmRec,arrStockAlmFin);
    emitirReporte(fechaIni,fechaFin,arrStockProd,arrStockAlm,cantStockProdAlm,
            arrProdAlmStockIni,arrProdAlmIng,arrProdAlmSal,arrProdAlmEnv,
            arrProdAlmRec,arrStockAlmFin,"ReporteProductosTotal.txt");

    emitirReporteProductosSeleccionados(fechaIni,fechaFin,arrStockProd,arrStockAlm,cantSt
    ockProdAlm,
            arrProdAlmStockIni,arrProdAlmIng,arrProdAlmSal,arrProdAlmEnv,
            arrProdAlmRec,arrStockAlmFin,"ReporteProductosSeleccionados.txt");
    return 0;
}



/*
 * File:   funciones.h
 * Author: Silvia Vargas Cáceres
 *
 * Created on 08 de octubre de 2024, 02:54 PM
 */

#ifndef FUNCIONES_H
#define FUNCIONES_H

void cargarArrStockProductos(int *,int *,double *,int &);
void mostrarArrStockProductos(int *,int *,double *,int );
void ingresarFechas(int &,int &,int &,int &,int &);
int agruparFecha(int ,int ,int );
void procesarTransaccionesAlmProd(int ,int ,int *,int *,int ,double *,double *,
```

```
64          double *,double *);
65  void actualizarTransaccionProdAlm(int ,int ,char ,double ,int ,int *,int *,int ,
66          double *,double *,double *,double *);
67  int buscarAlmProd(int ,int ,int *,int *,int );
68  void calcularStockFinalAlmProd(int ,double *,double *,double *,double *,
69          double *,double *);
70  void ordenarProductos(int *,int *,int ,double *,double *,double *,double *,
71          double *,double *);
72  void cambiarInt(int &,int &);
73  void cambiarDouble(double &,double &);
74  void emitirReporte(int ,int ,int *,int *,int ,double *,double *,double *,double *,
75          double *,double *,const char*);
76  void imprimirCabecera(ofstream &,int ,int );
77  void desagruparFecha(int ,int &,int &,int &);
78  void imprimirFecha(ofstream &,int ,int ,int );
79  void emitirReporteProductosSeleccionados(int ,int ,int *,int *,int ,double *,
80          double *,double *,double *,double *,double *,const char *);
81  void imprimirLinea(ofstream &,char ,int );
82  #endif /* FUNCIONES_H */
83
84
85  /*
86   * File:   funciones.cpp
87   * Author: Silvia Vargas Cáceres
88   *
89   *Created on 08 de octubre de 2024, 02:54 PM
90   */
91
92  #include <iostream>
93  #include <fstream>
94  #include <iomanip>
95
96  using namespace std;
97
98  #include "funciones.h"
99
100 #define NO_ENCONTRADO -1
101 #define TAM_LINEA 130
102
103 void cargarArrStockProductos(int *arrStockProd,int *arrStockAlm,double
    *arrProdAlmStockIni,
104         int &cantStockProd){
105     ifstream archStockProd("stockProductos.txt",ios::in);
106     if (not archStockProd.is_open()){
107         cout<<"El archivo stockProductos.txt no se pudo abrir"<<endl;
108         exit(1);
109     }
110     int codProd,codAlm;
111     double stock;
112     cantStockProd=0;
113     while(true){
114         archStockProd>>codProd;
115         if (archStockProd.eof()) break;
116         archStockProd>>codAlm>>stock;
117         arrStockProd[cantStockProd]=codProd;
118         arrStockAlm[cantStockProd]=codAlm;
119         arrProdAlmStockIni[cantStockProd]=stock;
120         cantStockProd++;
121     }
122  }
123
124 void mostrarArrStockProductos(int *arrStockProd,int *arrStockAlm,double
    *arrProdAlmStockIni,
125         int cantStockProd){
126     ofstream archReporte("reporteStockProductos.txt",ios::out);
127     if (not archReporte.is_open()){
128         cout<<"El archivo reporteStockProductos.txt no se pudo abrir"<<endl;
129         exit(1);
130     }
```

```
131       archReporte<<fixed<<setprecision(2);
132       for (int i=0;i<cantStockProd;i++)
133           archReporte<<setw(12)<<arrStockProd[i]<<setw(12)<<arrStockAlm[i]<<
134                   setw(12)<<arrProdAlmStockIni[i]<<endl;
135   }
136
137   void ingresarFechas(int &ddIni,int &mmIni,int &aaIni,int &ddFin,int &mmFin,int &aaFin){
138       char car;
139       cout<<"Ingrese la fecha de fin (dia/mes/anio): ";
140       cin>>ddIni>>car>>mmIni>>car>>aaIni;
141       cout<<"Ingrese la fecha de inicio (dia/mes/anio): ";
142       cin>>ddFin>>car>>mmFin>>car>>aaFin;
143   }
144
145   int agruparFecha(int dd,int mm,int aa){
146      return aa*10000+mm*100+dd;
147   }
148
149   void procesarTransaccionesAlmProd(int fechaIni,int fechaFin,int *arrStockProd,
150           int *arrStockAlm,int cantStockProdAlm,double *arrProdAlmIng,double
          *arrProdAlmSal,
151           double *arrProdAlmEnv,double *arrProdAlmRec){
152       ifstream archTransacciones("transacciones.txt",ios::in);
153       if (not archTransacciones.is_open()){
154           cout<<"El archivo transacciones.txt no se pudo abrir"<<endl;
155           exit(1);
156       }
157       int alm,dd,mm,aa,hh,min,ss,prod,almAdic,fecha;
158       double cant;
159       char tipoMov,car;
160       while(true){
161           archTransacciones>>alm;
162           if (archTransacciones.eof()) break;
163           archTransacciones>>dd>>car>>mm>>car>>aa;
164           fecha=agruparFecha(dd,mm,aa);
165           if (fecha>=fechaIni and fecha<=fechaFin){
166               while(true){
167                   archTransacciones>>hh>>car>>min>>car>>ss>>prod>>cant>>tipoMov;
168                   if (tipoMov=='T')
169                       archTransacciones>>almAdic;
170                   actualizarTransaccionProdAlm(alm,prod,tipoMov,cant,almAdic,arrStockProd,
171                           arrStockAlm,cantStockProdAlm,arrProdAlmIng,arrProdAlmSal,
172                           arrProdAlmEnv,arrProdAlmRec);
173                   if (archTransacciones.get()=='\n') break;
174               }
175           }
176           else
177               while(archTransacciones.get()!='\n');
178       }
179   }
180
181   void actualizarTransaccionProdAlm(int alm,int prod,char tipoMov,double cant,int almAdic,
182           int *arrStockProd,int *arrStockAlm,int cantStockProdAlm,double *arrProdAlmIng,
183           double *arrProdAlmSal,double *arrProdAlmEnv,double *arrProdAlmRec){
184       int pos=buscarAlmProd(alm,prod,arrStockProd,arrStockAlm,cantStockProdAlm);
185       if (pos!=NO_ENCONTRADO){
186           if (tipoMov=='I')
187               arrProdAlmIng[pos]+=cant;
188           else
189               if (tipoMov=='S')
190                   arrProdAlmSal[pos]+=cant;
191               else{
192                   int
                      posAlmAdic=buscarAlmProd(almAdic,prod,arrStockProd,arrStockAlm,cantStockP
                      rodAlm);
193                   if (posAlmAdic!=NO_ENCONTRADO){
194                      arrProdAlmRec[posAlmAdic]+=cant;
195                      arrProdAlmEnv[pos]+=cant;
196                   }
```

```
            }
        }
    }

    int buscarAlmProd(int alm,int prod,int *arrStockProd,int *arrStockAlm,int cant){
        for (int i=0;i<cant;i++)
            if (arrStockProd[i]==prod and arrStockAlm[i]==alm) return i;
        return NO_ENCONTRADO;
    }

    void calcularStockFinalAlmProd(int cantStockProdAlm,double *arrProdAlmStockIni,
            double *arrProdAlmIng,double *arrProdAlmSal,double *arrProdAlmEnv,
            double *arrProdAlmRec,double *arrStockAlmFin){
        for (int i=0;i<cantStockProdAlm;i++)
            arrStockAlmFin[i]=arrProdAlmStockIni[i]+arrProdAlmIng[i]-arrProdAlmSal[i]+
                    arrProdAlmRec[i]-arrProdAlmEnv[i];
    }

    void ordenarProductos(int *arrStockProd,int *arrStockAlm,int cantStockProdAlm,
            double *arrProdAlmStockIni,double *arrProdAlmIng,double *arrProdAlmSal,
            double *arrProdAlmEnv,double *arrProdAlmRec,double *arrStockAlmFin){
        for (int i=0;i<cantStockProdAlm-1;i++)
            for (int j=i+1;j<cantStockProdAlm;j++)
                if (arrStockProd[i]>arrStockProd[j] or
                    (arrStockProd[i]==arrStockProd[j] and
                    arrStockAlmFin[i]<arrStockAlmFin[j])){
                    cambiarInt(arrStockProd[i],arrStockProd[j]);
                    cambiarInt(arrStockAlm[i],arrStockAlm[j]);
                    cambiarDouble(arrProdAlmStockIni[i],arrProdAlmStockIni[j]);
                    cambiarDouble(arrProdAlmIng[i],arrProdAlmIng[j]);
                    cambiarDouble(arrProdAlmSal[i],arrProdAlmSal[j]);
                    cambiarDouble(arrProdAlmEnv[i],arrProdAlmEnv[j]);
                    cambiarDouble(arrProdAlmRec[i],arrProdAlmRec[j]);
                    cambiarDouble(arrStockAlmFin[i],arrStockAlmFin[j]);
                }
    }

    void cambiarInt(int &datoI,int &datoJ){
        int aux=datoI;
        datoI=datoJ;
        datoJ=aux;
    }

    void cambiarDouble(double &datoI,double &datoJ){
        double aux=datoI;
        datoI=datoJ;
        datoJ=aux;
    }

    void emitirReporte(int fechaIni,int fechaFin,int *arrStockProd,int *arrStockAlm,
            int cantStockProdAlm,double *arrProdAlmStockIni,
            double *arrProdAlmIng,double *arrProdAlmSal,double *arrProdAlmEnv,
            double *arrProdAlmRec,double *arrStockAlmFin,const char *nombReporte){
        ofstream archReporte(nombReporte,ios::out);
        if (not archReporte.is_open()){
            cout<<"El archivo "<<nombReporte<<" no se pudo abrir"<<endl;
            exit(1);
        }
        archReporte<<fixed<<setprecision(2);
        imprimirCabecera(archReporte,fechaIni,fechaFin);
        int i=0,prodCab;
        while(true){
            prodCab=arrStockProd[i];
            //almCab=arrStockAlm[i];
            imprimirLinea(archReporte,'=',TAM_LINEA);
            archReporte<<"PRODUCTO: "<<arrStockProd[i]<<endl;
            //archReporte<<"ALMACEN: "<<arrStockAlm[i]<<endl;
            imprimirLinea(archReporte,'-',TAM_LINEA);
            archReporte<<"ALMACEN"<<setw(20)<<"STOCK INICIAL"<<setw(12)<<"INGRESOS"
```

```
265                 <<setw(15)<<"SALIDAS"<<setw(22)<<"ENVIADO A ALMACENES"<<setw(22)<<
266                 "RECIBIDO DE ALMACENES"<<setw(14)<<"STOCK
                    FINAL"<<setw(14)<<"OBSERVACION"<<endl;
267         imprimirLinea(archReporte,'-',TAM_LINEA);
268         int j=i;
269         while(true){
270             archReporte<<setw(8)<<arrStockAlm[j]<<setw(15)<<arrProdAlmStockIni[j];
271             archReporte<<setw(15)<<arrProdAlmIng[j]<<setw(16)<<arrProdAlmSal[j];
272             archReporte<<setw(15)<<arrProdAlmEnv[j]<<setw(22)<<arrProdAlmRec[j];
273             archReporte<<setw(20)<<arrStockAlmFin[j];
274             if (arrStockAlmFin[j]<0)
275                 archReporte<<setw(20)<<"Trans.Incorrectas";
276             archReporte<<endl;
277             if (j==cantStockProdAlm-1) {
278                 i=j;
279                 break;
280             }
281             j++;
282             if (prodCab!=arrStockProd[j]){
283                 i=j;
284                 break;
285             }
286         }
287         if (j==cantStockProdAlm-1) break;
288     }
289 }
290
291 void imprimirCabecera(ofstream &archReporte,int fechaIni,int fechaFin){
292     int dd,mm,aa;
293     archReporte<<setw(85)<<"CONSOLIDADO DE STOCKS POR PRODUCTO"<<endl;
294     archReporte<<setw(55)<<"DEL ";
295     desagruparFecha(fechaIni,dd,mm,aa);
296     imprimirFecha(archReporte,dd,mm,aa);
297     archReporte<<setw(5)<<"AL ";
298     desagruparFecha(fechaFin,dd,mm,aa);
299     imprimirFecha(archReporte,dd,mm,aa);
300     archReporte<<endl;
301 }
302
303 void desagruparFecha(int fecha,int &dd,int &mm,int &aa){
304     dd=fecha%100;
305     aa=fecha/10000;
306     mm=(fecha%1000)%100;
307 }
308
309 void imprimirFecha(ofstream &archivo,int dd,int mm,int aa){
310     archivo<<setfill('0')<<setw(2)<<dd<<'/'<<setw(2)<<mm<<'/'<<setw(4)<<aa<<setfill(' ');
311 }
312
313 void emitirReporteProductosSeleccionados(int fechaIni,int fechaFin,int *arrStockProd,
314         int *arrStockAlm,int cantStockProdAlm,double *arrProdAlmStockIni,
315         double *arrProdAlmIng,double *arrProdAlmSal,double *arrProdAlmEnv,
316         double *arrProdAlmRec,double *arrStockAlmFin,const char *nombReporte){
317     double stockSel;
318     cout<<"Ingrese el stock mínimo para seleccionar los productos: ";
319     cin>>stockSel;
320     for (int i=cantStockProdAlm-1;i>=0;i--)
321         if (arrStockAlmFin[i]<=stockSel){
322             for (int j=i;j<cantStockProdAlm-1;j++){
323                 arrStockProd[j]=arrStockProd[j+1];
324                 arrStockAlm[j]=arrStockAlm[j+1];
325                 arrProdAlmStockIni[j]=arrProdAlmStockIni[j+1];
326                 arrProdAlmIng[j]=arrProdAlmIng[j+1];
327                 arrProdAlmSal[j]=arrProdAlmSal[j+1];
328                 arrProdAlmEnv[j]=arrProdAlmEnv[j+1];
329                 arrProdAlmRec[j]=arrProdAlmRec[j+1];
330                 arrStockAlmFin[j]=arrStockAlmFin[j+1];
331             }
332             cantStockProdAlm--;
```

```
333         }
334         emitirReporte(fechaIni,fechaFin,arrStockProd,arrStockAlm,cantStockProdAlm,
335                 arrProdAlmStockIni,arrProdAlmIng,arrProdAlmSal,arrProdAlmEnv,
336                 arrProdAlmRec,arrStockAlmFin,nombReporte);
337     }
338 
339     void imprimirLinea(ofstream &archivo,char car,int cant){
340         for (int i=0;i<cant;i++)
341             archivo.put(car);
342         archivo<<endl;
343     }
344 
```