***MODELO ABM***

#include <stdio.h>

#include <conio.h>

#include <stdlib.h>

struct producto

{

    int codigo,stock,borrado;

};

void altas(FILE \*);

void bajas(FILE \*);

void modificacion(FILE \*);

void compactar(FILE \*\*);

void listado(FILE \*);

void buscar(FILE \*\*,int);

void listadordenado (FILE\*);

void ordenar(struct producto\*, int);

int menu();

**int main()**

{

    FILE \*arch;

    int op;

  arch=fopen("PRODUCTOS.DAT","ab+");

    if(arch==NULL)

    {

        printf("No se pudo abrir el archivo\n");

        exit(-1);

    }

    fclose(arch);

    arch=fopen("PRODUCTOS.DAT","rb+");

    if(arch==NULL)

    {

        printf("No se pudo abrir el archivo\n");

        exit(-1);

    }

    op=menu();

    while(op!=7)

    {

        switch(op)

        {

            case 1: altas(arch);

                    break;

            case 2: bajas(arch);

                    break;

            case 3: modificacion(arch);

                    break;

            case 4: compactar(&arch);

                    break;

            case 5: listado(arch);

                  break;

           case 6: listadordenado(arch);

        }

        getch();

        op=menu();

    }

    fclose(arch);

    return 0;

}

**void altas(FILE \* pa)**

{

   int cod;

   struct producto reg;

   clearerr(pa);

   fseek(pa,0,SEEK\_SET);

   printf("Ingrese el codigo de producto 0 para terminar: ");

   scanf("%d",&cod);

   while(cod!=0)

   {

       buscar(&(pa),cod);

       if(!feof(pa))

            printf("Alta existente\n");

       else

       {

            clearerr(pa);

            reg.codigo=cod;

            reg.borrado=0;

            printf("Ingrese el stock: ");

            scanf("%d",&reg.stock);

            fseek(pa,0,SEEK\_END);

            fwrite(&reg,sizeof(struct producto),1,pa);

       }

       printf("Ingrese otro código 0 para terminar: ");

       scanf("%d",&cod);

   }

}

**void bajas(FILE \*pa)**

{

   int cod;

   struct producto reg;

   clearerr(pa);

   fseek(pa,0,SEEK\_SET);

   printf("Ingrese el código de producto que quiere eliminar: ");

   scanf("%d",&cod);

   while(cod!=0)

   {

       buscar(&(pa),cod);

       if(!feof(pa))

       {

           fseek(pa,ftell(pa)-sizeof(struct producto),SEEK\_SET);

           fread(&reg,sizeof(struct producto),1,pa);

           reg.borrado=1;

           fseek(pa,ftell(pa)-sizeof(struct producto),SEEK\_SET);

           fwrite(&reg,sizeof(struct producto),1,pa);

       }

       else

       {

           printf("Baja inexistente\n");

       }

       printf("Ingrese otro código de producto 0 para terminar: ");

       scanf("%d",&cod);

   }

}

**void modificacion(FILE \*pa)**

{

   int cod,stock;

   struct producto reg;

   clearerr(pa);

   fseek(pa,0,SEEK\_SET);

   printf("Ingrese el código de producto que quiere modificar: ");

   scanf("%d",&cod);

   while(cod!=0)

   {

       buscar(&(pa),cod);

       if(!feof(pa))

       {

           fseek(pa,ftell(pa)-sizeof(struct producto),SEEK\_SET);

           fread(&reg,sizeof(struct producto),1,pa);

           printf("Ingrese el nuevo stock: ");

           scanf("%d",&stock);

           reg.stock=stock;

           fseek(pa,ftell(pa)-sizeof(struct producto),SEEK\_SET);

           fwrite(&reg,sizeof(struct producto),1,pa);

       }

       else

       {

           printf("Modificación inexistente\n");

       }

       printf("Ingrese otro código de producto 0 para terminar: ");

       scanf("%d",&cod);

   }

}

**void compactar(FILE \*\*pa)**

{

   struct producto reg;

   FILE \*ftemp;

   ftemp=fopen("PRODUCTOS.TMP","wb");

   if(ftemp==NULL)

   {

       printf("No se pudo abrir el archivo\n");

       exit(-1);

   }

   clearerr(\*pa);

   fseek(\*pa,0,SEEK\_SET);

   fread(&reg,sizeof(struct producto),1,\*pa);

   while(!feof(\*pa))

   {

      if(reg.borrado==0)

         fwrite(&reg,sizeof(struct producto),1,ftemp);

      fread(&reg,sizeof(struct producto),1,\*pa);

   }

   fclose(ftemp);

   fclose(\*pa);

   remove("PRODUCTOS.DAT");

   rename("PRODUCTOS.TMP","PRODUCTOS.DAT");

   \*pa=fopen("PRODUCTOS.DAT","rb+");

   if(\*pa==NULL)

   {

       printf("No se pudo abrir el archivo\n");

       exit(-1);

   }

}

**void listado(FILE \* pa)**

{

  struct producto reg;

  clearerr(pa);

  fseek(pa,0,SEEK\_SET);

  fread(&reg,sizeof(struct producto),1,pa);

  while(!feof(pa))

  {

      printf("Código: %d Stock: %d Borrado: %d\n",reg.codigo,reg.stock,reg.borrado);

      fread(&reg,sizeof(struct producto),1,pa);

  }

}

**int menu()**

{

    int op;

    system("cls");

    printf("1- Alta\n");

    printf("2- Baja\n");

    printf("3- Modificación\n");

    printf("4- Compactar\n");

    printf("5- Listado\n");

    printf("6- Listado Ordenado\n");

   printf("7- salir\n");

    printf("\nIngrese una opción: ");

    do

    {

        scanf("%d",&op);

        if(op<1||op>7)

        {

            printf("Opción incorrecta. Intente de nuevo.\n");

        }

    }while(op<1||op>7);

    return op;

}

**void buscar(FILE \*\*pa,int c)**

{

  struct producto reg;

  clearerr(\*pa);

  fseek(\*pa,0,SEEK\_SET);

  fread(&reg,sizeof(struct producto),1,\*pa);

  while(!feof(\*pa)&&reg.codigo!=c)

  {

      fread(&reg,sizeof(struct producto),1,\*pa);

  }

}

**void listadordenado (FILE\*pa)**

{

  struct producto \*v, reg;

 int tam, i=0;

 clearerr (pa);

 fseek(pa,0,SEEK\_END);

tam=ftell(pa)/sizeof(struct producto);

v=(struct producto\*) malloc (sizeof(struct producto)\*tam);

if (v==NULL)

{    printf ("memoria insufuecuente");

    exit (-1);

}

clearerr (pa);

fseek(pa,0,SEEK\_SET);

fread(&reg,sizeof(struct producto),1,pa);

while (!feof(pa));

{

\*(v+i)=reg;

i++;

fread (&reg,sizeof(struct producto),1,pa);

}

ordenar(v,tam);

for (i=0;i<tam;i++)

printf ("Còdigo: %d  stock %d\n", (v+i)->codigo, (v+i)->stock);

free(v);

}

void ordenar (struct producto\*v, int n)

{

int p, c;

struct producto aux;

for (p=1;p<n;p++)

for(c=0;c<n-p;c++)

if ((v+c)->codigo>(v+c+1)->codigo)

{

 aux=\*(v+c+1)=aux;

}

}