

PRÀCTICA 2:GESTIÓ DE MAGATZEM

Em de realitzar un algoritme que sigui capaç de dur a terme les demandes i les exigències d'un magatzem que em creat nosaltres a la nostra manera .

En el nostre algoritme i tenim les següents estructures : primer tenim una tupla referida als productes(tconjprod), dins la tupla hi ha un camp enter el qual es refereix a la dimensió usada de la següent taula , també hi apareix dins la tupla una taula on hi ha els productes , aquesta taula esta delimitada a un màxim (60) el qual ens indica la dimensió declarada.

Cada producte està compost per un codi(el qual és una taula de quatre caràcters) ,un numero d'unitats disponibles, numero mínim d'unitats i un numero d'unitats venudes.

Respecte a les unitats disponibles , són les unitats que hi ha en stock en aquell moment en el magatzem. Respecte el numero d'unitats mínimes s'ha de dir que seran les unitats que s'hauran de demanar com a mínim per poder ser servit si les unitats demanades .

Les unitats venudes ens serviran per anar sumant les unitats de productes que siguin servides , i al final a través de l'estadística obtindrem els tres productes més venuts i els tres productes menys venuts.

Pel que fa respecte als clients, tindrà una estructura força semblant a la dels productes .Tindrem una tupla(tconclie) on hi apareixerà una taula amb els clients (tcclie)i un altre camp on ens indicarà la dimensió usada de la taula(n).La taula tindrà un màxim establert(60) el qual ens indicarà la dimensió declarada .Dins la taula hi ha els productes , i cada producte està format per una tupla on hi ha el nom i un contador (tsumat) .El nom està compost per tres caràcters , que correspondran a les tres inicials de la persona .La taula tsumat és una taula on va sumant les unitats que demana el client ,de cada producte diferent , si aquestes unitats no superen al mínim establert el que farà serà anar sumant les unitats .Un cop arribat al mínim el que farà es servir als clients i per això voldrà dir que el tsumat s'igualarà a zero .

Un cop es pugui servir el que farà és escriure el nom del client el codi del producte demanat i les unitats que s'han demanat .

Un cop haguim acabat de servir les comandes ,el que farà és fer una estadística dels tres productes més venuts i dels tres productes menys venuts.

Caldrà tenir en compte algunes consideracions : per servir les unitats demanades pel client hauràn de superar el mínim establert pel producte i les unitats demanades no hauran de superar les unitats disponibles en stock si tot això es compleix després es podrà servir.

S'han de partir d'unes hipotesis :

- Al principi coneixem el numero de clients que tindrem , i el nom de cadascun d'ells.
- El nom del client està compost per tres caràcters , el primer caràcter és refeix a la primera inicial del nom , el segon caràcter es refereix a la primera inicial del cognom i el tercer caràcter es refereix a la primera inicial del segon cognom , per tant les repeticions no seran gaire probables.
- Cada client pot fer diverses demandes i pot demanar diversos productes
- El codi de cada producte està constituït per quatre caràcters .

Esquematzació de les estructures anteriors:

ALGORITME

Programa del magatzem

Alg

Tipus

Tconjprod tupla :

d:enter

prod:tcproducte:

ftupla

tcproducte=taula [1..60]de tproduc

tproduc tupla:

codi : tcodi nd: enter nm : enter nv : enter; ftupla

tcodi = taula [1...4] de caràcters

tconjclie tupla :

clie: tcclie

n : enter

ftupla

tcclie = taula [1...60] de client

client tupla :

nom : tnom

sumat : tsumat

ftupla

tsumat = taula [1...60] de enter

tnom = taula [1..3] de caràcters

variables coclie : tconjclie coprod : tconjprod

res : eneter

fvariables

llegircp(coprod)llegircc(coclie)

{Escriure : Queden clients encara ?? \n OPCIONS: \n 1-No queden clients \n 2-Queden clients \n Quina opció esculls?}

res:=llegirEnter() mentres (res=2) fer

alvará(coprod,coclie; {Escriure : Queden clients encara ?? \n OPCIONS: \n 1-No queden clients \n 2-

Queden clients \n Quina opció esculls?} res:=llegirEnter());

fmentre

si (res=1) llavors biut

{escriure : Fi de Programa}

fsi

si no (res =1)i no (res=2) fer

{Escriure : Opció no reconeguda }

abortar

fsi

obtenir (coprod)

falgoritme

Definició de seqüència

Element de la seqüència alvará

Com obtenir el primer A través de la funció res:= llegirEnter() posada a fora el mentre

Com obtenir següent A través de la funció res:=llegirEnter() posada a dins el mentre i el que fa és anar llegint el res

Fi de seqüència Amb una condició establerta que res =2

Es un esquema de cerca ja que quan troba un res que no és 2, el que fa és parar i no continua fins al final,per tant es tracta d'una estructura de cerca .

IMPLEMENTACIONSacció llegircp(sort a tconjprod)

var

i,j: enter

fvar

{Escriure :Escriu el nombre de productes que vols tenir al magatzem} a.d:=llegirEnter()

per j de 1 a d fer

{Escriure : Escriu un codi de quatre caracters per cada producte:} per i de 1 a 4 fer
a.prod[j].codi[i]:=llegircaracter()

fper

{Escriure : Escriure el nombre disponible de productes al magatzem} a.prod[j].nd:=llegirenter() {Escriure :
Escriu el nombre minim d'unitats que s'han de comprar per

poder servir} a.prod[j].nm:=llegirenter() a.prod[j].nv:=0fper

facció /*****/ acció llegircc
(sort b tconjclie) var k,l : enter

fvar

{Escriure : Escriu el nombre de clients que vols tenir:}

b.n:=llegirenter() per k de 1 a b.n fer

{Escriure : Escriu les tres primeres inicials del client (1era inicial la del

nom ,2ona inicial la del 1er cognom,3era inicial la del 2on cognom):} per l de 1 a 3 fer
b.clie[k].nom[l]:=llegircaracter() fper

b.clie[k].sumat [k]:=0 fper

facció

/***/acció alvara (ent/sort a
tconjprod ,ent/sort b tconjclie) var codi : tcodi

nom : tnom c,d,e,posicio,pose,npd : enter

fvar

{Escriure : Introdueixi el codi del producte desitjat(quatre caracters):} per c de 1 a 4 fer

codi[c]:=llegircaracter() fper

{Éscriure : Introdueixi el nombre d'unitats desitjades:} npd:=llegirenter() {Escriure : Introdueixi el nom del
client(tres caracters):} per d de 1 a 3 fer

nom[d]:=llegircaracter()

fper

relacionar(b,nom,posicio)servir(b,a,codi,nom,npd,pose,posicio)

facció

/******/acció relacionar
(ent/sort b tconjclie ,ent nom tnom ,sort posicio enter) var

z,m,o : enter

fvar

z:=1;m:=3;o:=1; mentres m=3 i z<b.n+1 m:=compararnom(b.clie[z].nom,nom) z=z+1fmentre si m=2 i z<b.n+1 fer

posicio:=z m=3 o z>b.n {Escriurà : El nom del client no existeix} buit

sino parar

fsi

facció

Definició de seqüència

Element de la seqüència El nom del client

Com obtenir el primer Primer inicialitza tenint en compte que el nom no correspon a cap client de la taula (m:=3) i després comença pel primer client i el compara amb el nom entrat (nom)

Com obtenir següent A través d'un contador que es dedica a passar al següent client

Fi de seqüència Amb una condició establerta que mentres m:=3 i amb una altre condició de fi que és que z no superi la dimensió usada dels clients (b.n)

Es un esquema de cerca ja que quan troba un nom d'un client igual al entrat a posteriori , m pren per valor 2 i per tant surt de la seqüència , ja que està fora de la condicions establertes.

/******/funció compararnom
(ent n1 : tnom, ent n2 tnom) retorna enter

var

o,m : enter

fvar

o:=1

mentre n1[o]=n2[o] i no o=4 fer

o:=o+1

fmentre

si o=4 llavors m:=2

o<5 llavors m:=3fsiretorna m

ffunció

Definició de seqüència

Element de la seqüència un parametre del tipus tnom

Com obtenir el primer El primer caràcter a comparar es pren per o:=1

Com obtenir següent A través d'un contador que va passant al següent caràcter, o:=o+1

Fi de seqüència Hi ha dues condicions una que és que es dedica a mirar si els caràcters són iguals i una altra que indica si ha arribat al últim caràcter , això implica que si ha arribat al últim caràcter els noms són iguals si no els noms són diferents i canvia de client.

Es un esquema de cerca ja que quan troba una seqüència de noms (serie de caràcters) i hi ha algún caràcter que no és igual o ja s'ha acabat el nom , para la seqüència.

/******/acció servir
(ent/sort c tconjclie,ent/sort h tconjprod,ent e tcodi ,ent f tnom, ent npd

enter, sort pose enter ,ent posicio enter)var

q,x,g,res : enter

fvar

res:=3q:=1mentres res=3 i q<h.(d+1) fer res:=comparacodi(h.prod[q].codi ,e)

q:=q+1fmentresi res=2 i q<h.(d+1) llavors pose:=q;q>h.(d+1)o res=3 llavors {Escriure: El codi del producte no existeix} buit

sino parar

fsi((c.clie[posicio].sumat[pose]):=c.clie[posicio].sumat[pose]+npd)si (no(
c.clie[posicio].sumat[pose]<h.prod[pose].nm) i no

(c.clie[posicio].sumat[pose]>h.prod[pose].nd)) llavors

(h.prod[pose].nv):=(h.prod[pose].nv)+(c.clie[posicio].sumat[pose])
h.prod[pose].nd:=(h.prod[pose].nd)-(c.clie[posicio].sumat[pose])

escriurenom(c,posicio)

escriurecodi(h,pose) {Escriure : El nombre d'unitats que han sigut servides han sigut }
escriureenter(c.clie[posicio].sumat[pose]);

c.clie[posicio].sumat[pose]:=0

(c.clie[posicio].sumat[pose]<h.prod[pose].nm) llavors

{Escriure : Els productes demanats no superen el limit establert per ser servir , el minim establert és:}

escriureenter(h.pro[pose].nm)

(c.clie[posicio].sumat[pose]>h.prod[pose].nd) llavors {Escriure : Els productes demanats no han sigut subministrats per falta de stock en el magatzem, els productes disponibles son:}

escriureenter(h.prod[pose].nd)sino llavors parar

fsi

facció

Definició de seqüència

Element de la seqüència un parametre del tipus tcodi

Com obtenir el primer El primer caràcter a comparar es pren per (q:=1) i s'ha de

tenir en compte que es parteix de res:=3 per tant que els

codis són diferents

Com obtenir següent A través d'un contador que va passant al següent caràcter

del codi,qo:=q+1

Fi de seqüència Amb una condició establerta que mentres res:=3 i amb una altre condició de fi que és que q no superi la dimensió usada dels productes(h.d)

Es un esquema de cerca ja que quan troba un codi d'un producte igual al entrat a posteriori , res pren per valor 2 i per tant surt de la seqüència , ja que està fora de la condicions establertes. Aquesta estructura és molt semblant a la usada en relacionar .

/******/funció

compararcodi(ent x1 tcodi,ent x2 tcodi) var

k,res : enter

fvar

res:=3 k:=1

mentre x1[k]=x2[k] i no k=5 fer

k:=k+1

fmentre

si k=4 llavors res:=2

k<4 llavors

res:=3f*fsi*

retorna res

ffunció

Definició de seqüència

Element de la seqüència un parametre del tipus t*codi*

Com obtenir el primer El primer caràcter a comparar es pren per k:=1

Com obtenir següent A través d'un contador que va passant al següent caràcter, k:=k+1

Fi de seqüència Hi ha dues condicions de fi una que és que es dedica a mirar si els caràcters són iguals i una altra que indica si ha arribat al últim caràcter del codi , això implica que si ha arribat al últim caràcter , els codis són iguals si no ,els codis són diferents i canvia de producte.

Es un esquema de cerca ja que quan troba una seqüència de codis (serie de caràcters) i hi ha algún caràcter que no és igual, para la seqüència.

/*********/ acció escriure nom
(sort f t*conjclie*,ent posicio enter) var

r : enter

fvar

per r de 1 a 3 fer

{Escriure : El nom del client servit es:} escriurecaracter(f.clie[posicio].nom[r])

fper

facció

/*********/ acció escriurecodi(sort
e :t*conjprod* ,ent pose: enter) var

t : enter

fvar

per t de 1 a 3 fer

{Escriure : El codi del producte servit es:} escriurecaracter(e.prod[pose].codi[t])fper

facció/*********/

acció obtenir (ent/sort c :t*conjprod*)

var

y,pm : enter

fvar

per y de 1 a c.(d-1) llavors

pm:=buscar (c,y,c.d,)

ordenar (c,y,pm)

fper

escriur(c,c.d)

facció

/******

funció buscar (ent c tconjprod ,ent y enter, ent z enter,)

var

t,p : enter

fvar

p :=y

per t de (y+1) a (z) llavors

si c.prod[t].nv<c.prod[p].nv llavors

p:=t

sino buit

fsi

fper

retorna p

ffunció

/******

acció ordenar (ent c tconjprod , ent y enter, ent pm enter)

var

q,w,e : enter

fvar

per q de 1 a 4 fer

c.prod[c.(d+1)].codi[q]:=c.prod[y].codi[q]

fper

per w de 1 a 4 fer

c.prod[y].codi[w]:=c.prod[pm].codi[w]

fper

per e de 1 a 4 fer

c.prod[pm].codi [e]:=c.prod[c.(d+1)].codi[e]

fper

facció

ESPECIFICACIONS

- Acció llegircp (sort a tconjprod)

{El que fa aquesta acció és emplenar la taula dels productes fins a la seva dimensió usada , cada producte té varios camps nm,nd,nv,codi }

- Acció llegircc(sort b tconjclients)

{En aquesta acció el que es fa és emplenar la taula dels clients, fins la dimensió usada , amb el nom de cadascun i sumat és 0 }

- Acció alvarà (ent/sort a tconjprod,ent/sort b tconjclie)

{El que fa en aquesta acció és rebre la comanda del client (nom , codi del producte desitgat , numero d'unitats) i compara el nom del client i el codi amb als entrats a posteriori .Si són correctes es serveix sino aborta }

- Acció relacionar

{El que passa en aquesta acció és que relaciona els noms dels clients del principi amb els noms dels clients de la comanda , si és correcte m:=2 i si el nom no és correcte, que no correspon a cap dels anteriors , m:=3 }

- Funció compararnom (ent n1 : tnom , ent n2 : tnom)retorna boolea

{El que fa en aquesta funció és comparar cada caràcter dels clients amb als caracters dels clients que fan la comanda }

- Acció servir (ent/sort c tconjclie,ent/sort h tconjprod,ent e tcodi ,ent f tnom, ent npd

enter, sort pose enter ,ent posicio enter)

{El que fa aquí es lo mateix que ha fet anteriorment amb el nom però ara amb el codi el qual té quatre caràcters i el nom en té tres. A més a més ens fa la comanda , que això implica que si el numero d'unitats desitjades del producte és més petit que el mínim establert pel producte no es podrà servir , això també passa amb el numero d'unitats disponibles del producte però al revés. Si aquestes condicions es compleixen es podrà servir que implica escriure el nom del client el codi del producte desitjat i les unitats servides .}

- Funció comparar codis (ent x1 tcodi , ent x2 tcodi)

{El que fa és comparar els codis que em entrat al principi amb el codi que ha sigut demanat en la comanda i si existeix el codi m:=2 i si no existeix m:=3, per tant és una comparació caràcter per caràcter }

- Acció escriure nom (sort f tconjlie, ent posicio enter)

{El que fa en aquesta acció és escriure caràcter per caràcter el nom del client , i aquest serà una senyal de que s'ha complert la comanda }

- Acció escriure codi (sort e : tconjprod, ent pose : enter)

{Escriu el codi del producte desitjat en la comanda i també és senyal de que la comanda s'ha realitzat sense cap problema }

- Obtenir (ent/sort c : tconjprod)

{El que fa en aquesta acció és buscar el valor mínim , ordenar la taula del productes de menys a més venuts i després escriu els tres més venuts i els tres menys venuts }

- Funció buscar (ent c : tconjprod, ent y enter , ent z enter)

{el que fa en aquesta acció és buscar la posició on hi ha el valor més petit , aquest valor està referit a les unitats venudes }

- Funció ordenar (ent c tconjprod , ent y enter , ent pm enter)

{