

OHARRAK: Azterketaren iraupena 3 ordutakoa da

Ez erantzun galderei orrialde honetan, banatuko diren folioetan baizik
Folioak zenbatu eta guzuzietan izena jarri

1 Galdere teorikoak (1.5)

- 1.1 (0.25) Pasa ezazu 110011 zenbaki biarra sistema hamartarra (adierazi nola kalkulatuzen duzun)
1.2 (0.25) Pasa ezazu 1151 zenbaki hamartarra sistema biarra (adierazi nola kalkulatuzen duzun)
1.3 (0.25) Zeintzuk dira PUZ baten barruan aurki daitezkeen oinarriko elementuak?
1.4 (0.5) Azaldu zer den konpladore bat, eta zertarako balio duen
1.5 (0.25) Zenbat MegaByte dira 2²⁵ bit?

2 C programazio lengoia erabiliz, ondorengo baldintzak idatzi (0.5)

- 2.1 (0.25) Idatzi n1, n2 eta n3 zenbakietako batek 5 balio duenean bakarrik betetzen den baldintza
2.2 (0.25) n1, n2 eta n3 zenbakiek balio berdina ote duen kontsultatuzen duen baldintza

3 (1) Zer egiten du ondoren azaltzen den kodeak?

```
#include <stdio.h>
void main()
int balioa;
int bider, kont;
printf ("Idatzi osoko zenbaki bat\n");
scanf ("%d", &balioa);
bider=1;
kont=1;
while (kont<=balioa)
{ bider=bider*2;
  kont=kont+1;
}
printf ("Kalkulatutako balioa %d da\n", bider);
}
```

4 (1) 20 osagatetako taula baten elementuen balioa pantalaratzen duen hurrengo kodea emanik, programazio erroreak zuzendu

```
.....
int taula [20]; pos;
.....
/*Taulako osagaien irakurketa*/
.....
/*Pantalaratzearen hasiera*/
for (pos=1; pos<=20; pos++)
{
  printf ("%6f", taula[pos]);
}
.....
```

5 (6) Ondoren azaltzen den artikerentzat C kodea eta fluxu diagrama idatzi.

Oharra: azpi-galdere bakoitzak 1.5 balio du. Puntu horietatik 0.5 fluxu diagramarentzat eta puntu bat C kodearentzat izango dira

Artiketa honek 10 ekipotako liga batetan jokatuzen diren partiduen emaitzak gordetzea eta aztertzea du helburu. Partidu bakoitzeko balio bakarra gordetzen da, balio hau ondoko bat izango da:

- 2 baldin eta etxean jokatuzen duen ekipok irabazi badu
- 1 baldin eta berdinketa bat gertatzen bada
- 0 etxean jokatuzen duen ekipok galdtu du
- -1 baldin eta partidua oraindik jokatu ez bada
- -2: partidua ezinezkoa da

Behin datuak sartuta taulak ondoko itxura izan dezake (adibide moduan)

Ekipotak	0 ek	1 ek	2 ek	3 ek	4 ek	5 ek	6 ek	7 ek	8 ek	9 ek
0 ekipoa	-2 ⁽¹⁾	2 ^(a)	1 ^(a)	0 ^(a)	-1 ^(s)	1	0	-1	1	0
1 ekipoa	0	-2	-1	2	2	-1	2	2	2	2
2 ekipoa	1	-1	-2	1	-1	0	2	1	2	2
3 ekipoa	-1	1	-1	-2	2	1	0	2	2	2
4 ekipoa	2	2	0	1	-2	2	-1	2	1	-1
5 ekipoa	-1	0	1	-1	0	-2	2	1	2	2
6 ekipoa	1	1	2	1	2	-1	-2	2	-1	2
7 ekipoa	2	-1	0	-1	2	2	1	-2	0	2
8 ekipoa	2	1	1	2	0	1	-1	0	-2	-1
9 ekipoa	2	2	2	0	1	2	0	-1	-1	-2

Beltzez dagoenak ondokoa azaltzen du:

- (1) 0. ekipok ezin du bere aurka jokatu, horregatik azaltzen da -2
(2) 0. ekipok bere etxean irabazi zuen 1go ekiporearen aurka
(3) 0. ekipok etxean berdindu zuen 2. ekiporearen aurka
(4) 0. ekipok etxean galdtu zuen 3. ekiporearen aurka. Edo bertina dena, 3. ekipok irabazi zuen 0. ekiporearen aurka 0. ekiporearen zelatari
(5) 0. ekipoa eta 4.aren arteko partidua, 0. ekiporearen zelatari oraindik ez da jokatu

Ondorengo arazoetarako erantzuten diren kodeak idatz ezazu:

- 5.1 90 partiduetako datuak irakurri eta aurrean azalduetako taularen moduko batean gorde. 90 partiduentzako erabilizailerik erakutsiko zaio:

- “Sartu ondoko partiduren emaitza: X ekipoa- Y ekipoa”, non X eta Y momentu horretan eskatzen ari garen partiduren ekipoa diren. Ondoren erabilzaileak partidu horretzako emaitza sartu beharke du (-1, 0, 1 edo 2)
- 5.2 Beste taula bat sortu, eta bertan gorde ekipa bakoitzak zenbat puntu lortu dituen bere zelaiari
 - 5.3 Beste taula bat sortu ekipa bakoitzaentzat gordeitzeko zenbat puntu lortu dituen guztira (etikoa eta beste zelaiariari jakatukoak)
 - 5.4 Puntuari taula orokorra ordenatu haunditik txikira

Oharrak:

- 2,3 eta 4 puntuak erantzun daitezke aurreko zatiake degoeneko ongi daudela suposatuz
- Partidu bakoitzekeo puntuak honela kalkulatu dira:
 - 3 baldin eta irabazia
 - 1 baldin eta berdinketa bat eman bada
 - 0 baldin eta galdu bada

2002/02/04 KO AZTERKETAREN EBAZPENA

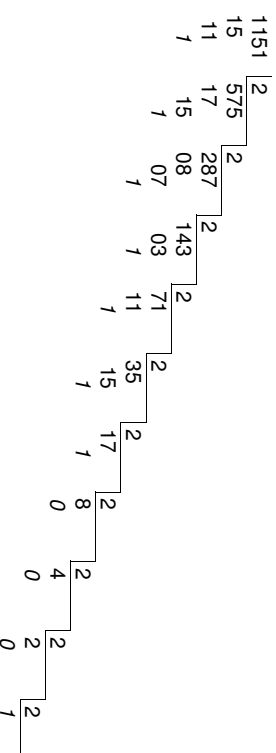
1go arriketa

1.1

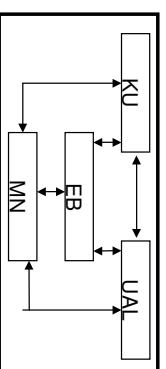
$$110011 = 2^5 + 2^4 + 2^1 + 2^0 = 32 + 16 + 3 + 1 = 51$$

1.2

$$1151 = 1000111111$$



1.3



1.4

Kompiadoreek goi-mailako programazio lengoai batean dagoen programa bat hartzen du eta makina lengoaiara itzultzen du.

1.5

2^{25} bit 4 Megabyte dira

2. arriketa

2.1

$$((n1 == 5) \parallel (n2 == 5) \parallel (n3 == 5))$$

2.2

$$((n1 == n2) \&\& (n1 == n3))$$

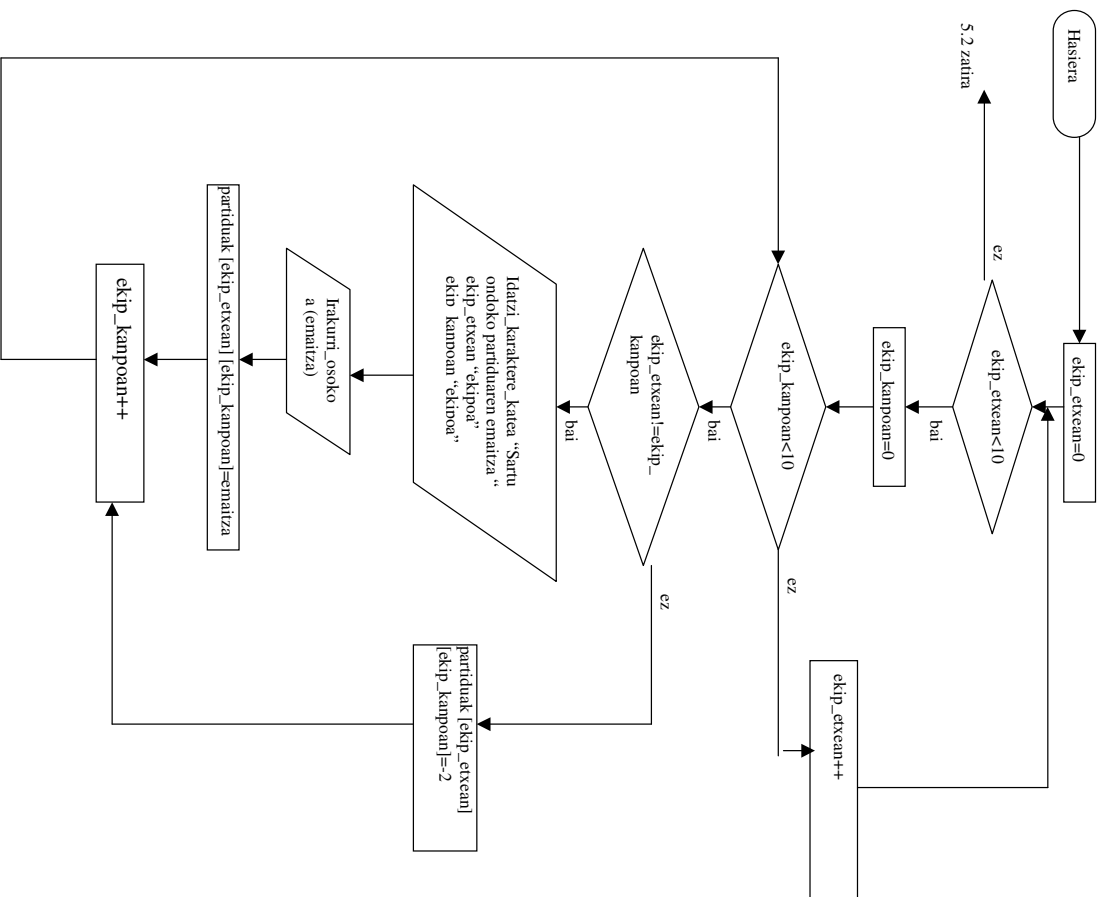
3. arriketa

Programak erabilzaileari zenbaki bat eskatzen dio (balioa) eta 2^{balio} pantailaratuko du

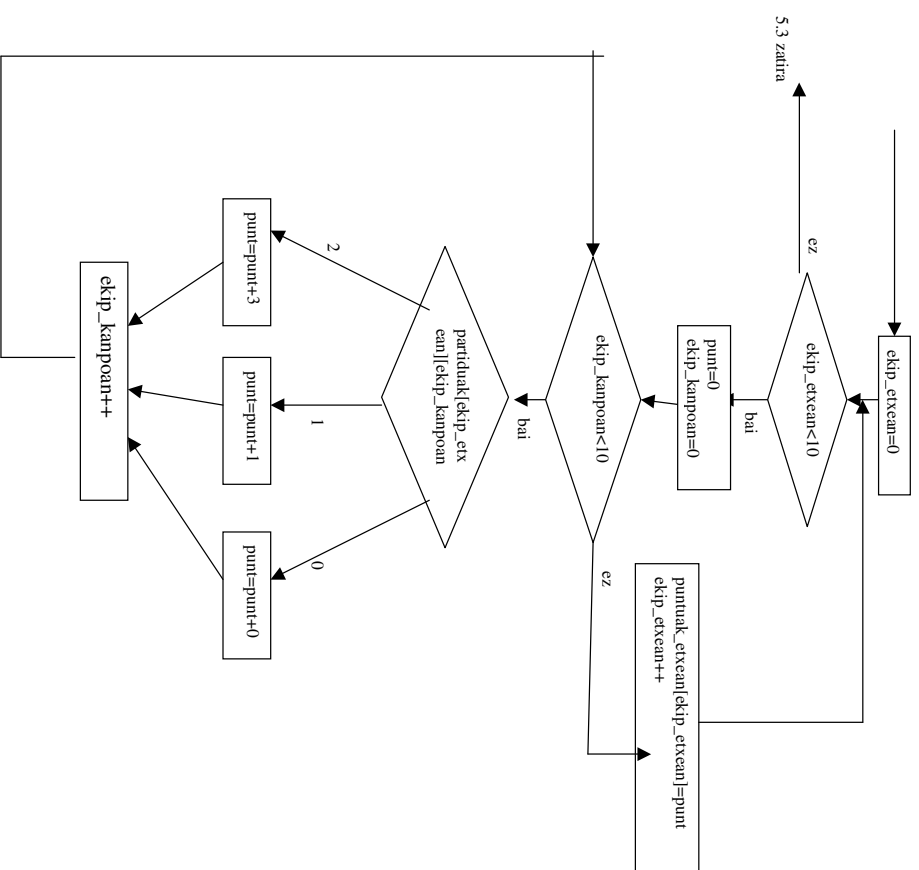
4. arriketa

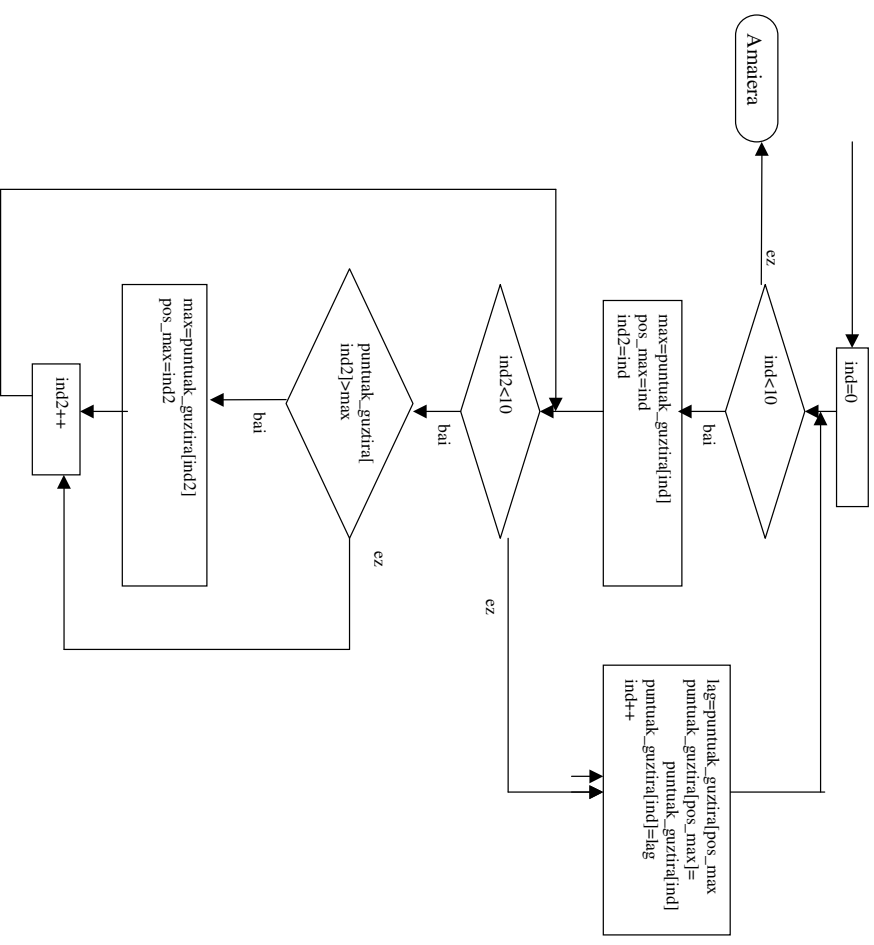
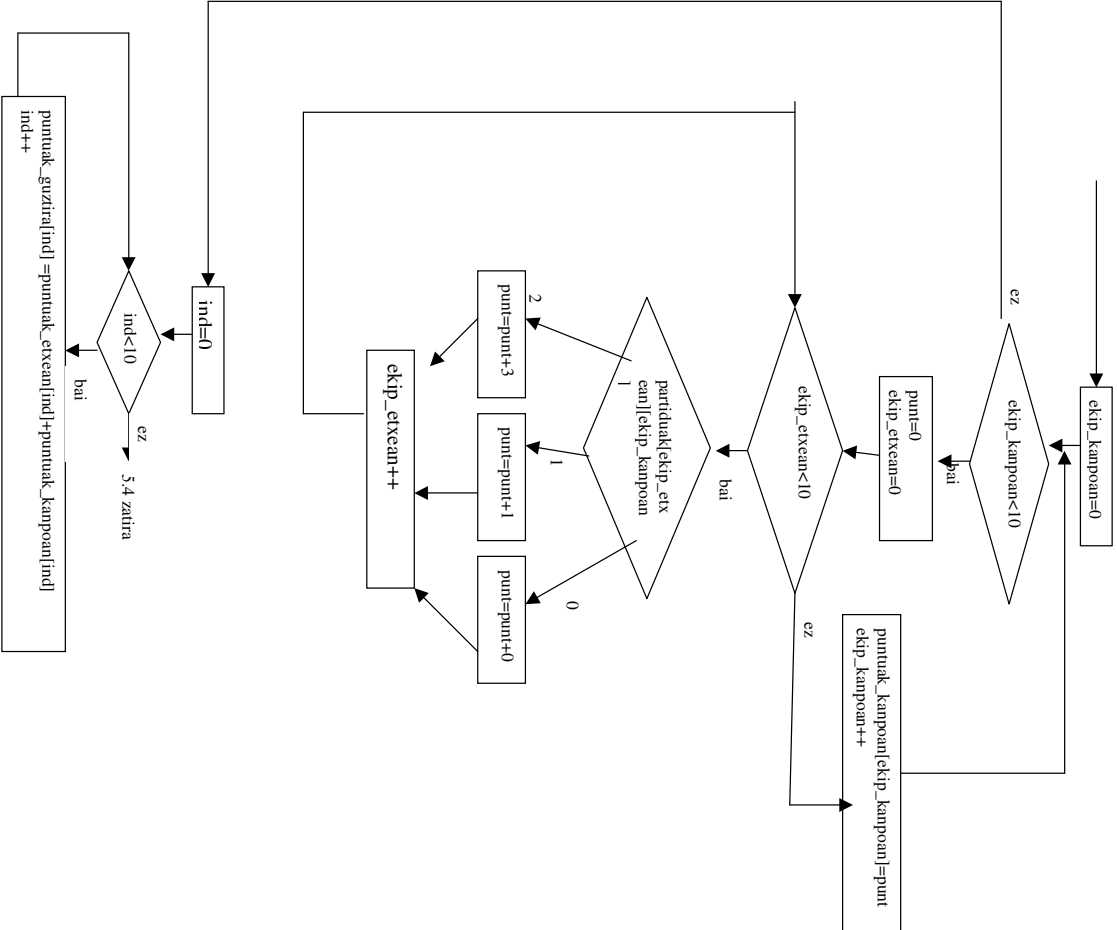
.....
 int taula [20] , pos;

5.1



5.2





5. arketeta (bigarren proposamena) C kodea

```
#include <stdio.h>
#include <conio.h>

void main ()
{
    int liga [10][10], etxekopuntuak[10], puntuak [10];
    int etxekoa, kanpokoa, emaitza, kont, i, aux, j;

    clrscr();
    for (etxekoa=0; etxekoa<=9; etxekoa++)
        for (kanpokoa=0; kanpokoa<=9; kanpokoa++)
            if (etxekoa == kanpokoa)
                [liga[etxekoa][kanpokoa]=-2;
            else
                {do{
                    printf ("\n%d ekipoa - %d ekiptaren kontra, zein da emaitza? ", etxekoa,
                    kanpokoa);

                    scanf ("%d" &emaitza);
                }while (emaitza>2 || emaitza <-1);
                liga[etxekoa][kanpokoa] = emaitza;
            }
        }
    puts ("      0 1 2 3 4 5 6 7 8 9");
    puts ("      -----");
    for (etxekoa=0; etxekoa<=9; etxekoa++)
        for (etxekoa=0; etxekoa<=9; etxekoa++)
            {printf ("%d ekipoa : ", etxekoa);
                for (kanpokoa=0; kanpokoa<=9; kanpokoa++)
                    {
                        printf ("%2d |", liga [etxekoa][kanpokoa]);
                        puts("\n      -----");
                    }
            }
    for (etxekoa = 0; etxekoa<=9; etxekoa++)
        {kont = 0;
            for (kanpokoa=9; kanpokoa>=0; kanpokoa--)
                {
                    switch(liga [etxekoa][kanpokoa])
                    {case 2: {kont = kont + 3;
                                break;}
                     case 1: {kont = kont +1;
                                break;}
                     }
                }
            etxekopuntuak[etxekoa] = kont;
        }
}
```

```
for (etxekoa = 0; etxekoa <=9; etxekoa++)
    {printf ("\n%d ekiptak %2d puntu", etxekoa, etxekopuntuak[etxekoa]);
    }
for (kanpokoa = 0; kanpokoa<=9; kanpokoa++)
    {kont = 0;
        for (etxekoa=9; etxekoa >=0; etxekoa--)
            {
                switch(liga [etxekoa][kanpokoa])
                {case 2: {kont = kont + 3;
                            break;}
                 case 1: {kont = kont + 1;
                            break;}
                 }
            }
        puntuak[kanpokoa] = kont;
    }
for (etxekoa=0; etxekoa <=9; etxekoa++)
    {puntuak[etxekoa] = puntuak[etxekoa] + etxekopuntuak[etxekoa];
        printf ("\n%d ekipoa : %d puntu totalen", etxekoa, puntuak [etxekoa]);
    }
getch();
for (i= 0; i < 10; i++)
    {for (etxekoa= 0; etxekoa < 9; etxekoa++)
        {
            if (puntuak[etxekoa] < puntuak [etxekoa+1])
                {aux=puntuak [etxekoa+1];
                    puntuak [etxekoa+1] = puntuak[etxekoa];
                    puntuak[etxekoa]=aux;
                }
            clrscr();
            for (j=0; j <=9; j++)
                {
                    printf ("%2d |", puntuak[j]);
                    delay(500);
                }
        }
    }
puts ("\nSakatu");
getch();
}
```

- 1 (0.25) Pasa ezazu 10110010001101 zenbaki biarra sistema hamartarera (adierazi nola kalkulatuzen duzun)
- 2 (0.25) Pasa ezazu 652 zenbaki hamartarra sistema bitarera (adierazi nola kalkulatuzen duzun)
- 3 (0.25) Idatz ezazu balintza bat ondorengo egoeran bakarrik betetzen dena: A, B eta C hiru osoko zenbaki bakoitiak direnean eta hamarrekoak berdinak dituztenean.
- 4 (0.25) Zenbat KByte dira 23456 bit?
- 5 (3) Idatzi C programa bat ondorengo egiten duena:
 - 5.1 Teklatuik karaktere kate bat irakurri eta s1 taula batean gorde
 - 5.2 do-while bat erabiltz kopiatu s1 s2 izeneko beste taula batean ondorengo ordezkapenak eginez:


```
a<<=m, e<<=j, i<<=i, o<<=c eta u<<=p
```
 - 5.3 Pantailaratu s1 eta s2 taulak
- 6 (5) Ondoren azaltzen den ariketarentzat C kodea eta fluxu diagrama idatzi.

Sukaldari batek 10 errezetako liburu bat dauka eta beste gauzen artean bakoitzak beharrezkoak dituen produktuak eta haien kantitateak gordetzen dira. Ondorengo galderei erantzuteko gai izango den programa bat egin:

 - 6.1 Irakurri eta taula batean gorde errezeta bakoitzak behar duen produktu kantitatea. Erabiltzen ez diren produktuak 0 bat izan beharko duze (ez dira zertan deklaratu irakurri behar). Irakurketa formatua honedokoa da:


```
Sartu produktu kantitateak:
Errezeta, produktua, kantitatea: 1 2 20
.....
Errezeta, produktua, kantitatea: 0 0 0
Hau da, irakurketa 0 0 0 irakurtzerakoan amaituko da.
```
 - 6.2 Esan zein den hirugarren produktuko kantitate gehien behar duen errezeta
 - 6.3 Taula berri bat sortu eta bertan errezeta bakoitzak behar duen produktu desberdin kopurua sartu.
 - 6.4 Ordenatu aurreko taula handianetik txikienera.
 - 6.5 Egin funtzio bat balio bat emanda Oren bertina den edo ez adieraziko duena. Funtzio hori erabiltz esan ea lehenengo errezetak zenbat produktu desberdin erabiltzen dituen.
- 7 (1) Hiru zenbaki irakurri eta handiena zein den erabakitzen duen programa baten fluxu diagrama egin. Behin eginda, 3 osoko balio aukeratu eta zure fluxu diagrama jarraituz exekuzio diagrama egin.

2003/01/20-A KO AZTERKETAREN 6. ARIKETAREN EBAZPENA

```
#include <stdio.h>
#include <string.h>
#include <conio.h>

//funtzioaren prototipoa
int desberdin0 (int);
//konstanteen deklarazioa
#define errezetak 10
#define produktuak 20
//programa nagusia
void main() {
    int er, pro, kan, max, max_pos, batura, ind1, ind2, lag;
    int taula[errezetak][produktuak], produk[errezetak];
    clrscr();

    //6.1 GALDERA
    //Hasieraketa
    for (er=0;er<errezetak;er++)
        {for (pro=0;pro<produktuak;pro++)
            {taula[er][pro]=0;}}

    //Balioen irakurketa
    printf ("datuak sartu. Errezeta, produktua, kantitatea:");
    scanf ("%d %d %d", &er, &pro, &kan);
    while (i((er==0) && (pro==0) && (kan==0)))
        {taula[er-1][pro-1]=kan;
        printf ("\n Errezeta, produktua, kantitatea:");
        scanf ("%d %d %d", &er, &pro, &kan);
        }

    //6.2 GALDERA
    max=taula[0][2];max_pos=0;
    for (er=1;er<errezetak;er++)
        {if (max<taula[er][2])
            {max=taula[er][2];
            max_pos=er;
            }
        }
    printf ("\n Gehien duena %d da \n", max_pos+1);

    //6.3 GALDERA
    for (er=0;er<errezetak;er++)
        {batura=0;
        for (pro=0;pro<produktuak;pro++)
            {if (taula[er][pro]i=0)
                {batura++;}
            }
        produk[er]=batura;
    }

    //6.4 GALDERA
    for (ind1=0; ind1<=errezetak; ind1++)
        {max=produk[ind1];max_pos=ind1;
        }
    }
```

```

for (ind2=ind1;ind2<errezetak;ind2++)
{ if (produk[ind2]>max)
{max=produk[ind2];
max_pos=ind2;
}
lag=produk[ind1];
produk[ind1]=produk[max_pos];
produk[max_pos]=lag;
}
}
//pantailaraztea
for (er=0;er<errezetak;er++)
{printf ("%d ",produk[er]);}
//6.5 GALDERA
batura=0;
for (pro=0;pro<produktuak;pro++)
{if (desberdin0(taula[0][pro]))
{batura++;}
}
printf ("\n %d", batura);
}
//Funtzioaren gorputza
int desberdin0 (int zenb) {
if (zenb==0)
return 0;
else
return 1;
}
}

```

Izen abizenak: 2003/01/20-B

OHARRAK: Azterketaren iraupena 3 ordukoa da

Ez erantzun galderei orrialde hauean, banatuko diren folioetan baizik
Folioak zenbatu eta guztieta izena jarri

1 Galdere teorikoak (1.25)

- 1.1 (0.25) Pasa ezazu 100101 zenbaki bitarra sistema hamartarra (adierazi nola kalkulatu den duzun)
- 1.2 (0.25) Pasa ezazu 71 zenbaki hamartarra sistema bitarrera (adierazi nola kalkulatu den duzun)
- 1.3 (0.25) Zer da eta zergatik sortzen dira nihizadura lengoaiak?
- 1.4 (0.25) Adierazi zein esaldi dauden ongi eta zeintzuk gaizki: Gaizki daudenak zuzendu
TurboPascalak diskaren arxibo sistemaren kudeatzailea da

Kompiadoreak sintaxi erroreak zuzentzen ditu

- 1.5 (0.25) 23456 bit zenbat KByte dira?

2 (1) Zer pantailarazten du ondoren azaltzen den kode zatiak?

3 (1) duen hurrengo kodea emanik, programazio erroreak zuzendu

```

include <stdio.h>
void main(){
int pos;
int a ; taul[pos];
int suma=0 ;
/lectura de elementos de la tabla
pos =10;
do
suma=suma+taul[pos] ;
printf ("%d", pos);
pos--;
while (pos<0)
printf (« d »>suma)
}

```

4 (2) Zenbaki bat emanda perfektua den edo ez (bere zatizailde guztien batura bera kenduta beraren berdina bada) adierazten duen funtzio batenzati fluxu diagrama eta programa egin itzazu.

Adibidez : 6 perfektua da $6=1+2+3$ delako
28 perfektua d $28=1+2+4+7+14$ delako

Egin programa bat lehenadabiziko hamar zenbaki perfektuak pantailarazteko aurreko funtzioa erabiliz.

5 (4) Ondoren azaltzen den arteketarantzai C kodea eta fluxu diagrama idatzi.

Hotel baten okupazioa kudeatzeko duen programa bat nahi dugu. Hotela 20 gela dauzka eta 15 egunerako datuak gorde nahi ditugu. Ondorengo galdereak erantzuteko gai izan behar du programak:

- a) Hotelaen okupazio datuak irakurri (denak eta ordenan irakurri)
- b) Guztiz betetako gela kopurua (2 pertsona), partzialki betetakoena (pertsona bat) eta hutsena kalkulatu
- c) Egun bat eta pertsona kopuru bat emanda esan ea egun horretan hotelaan lekurik ore dauden.
- d) Kalkulatu zenbat gela dauden okupatuta egun bakoitzean eta ondoren esan zein den gela libre gehien dauzkan egunak.

6 (1) Aurreko arazotaren egoera berdinean, datuak hirukotea sartzen dira (eguna, gela, pertsona-kopurua) ordenatu gabe eta gainera, gela hutsen datuak ez dira sartzen. Datu-sarretako sekzioa berriro ere idatzi aldaketa hauek kontutan izanik.

Aldibidea:

Okupazio datuak sartu:

Eguna, gela, pertsona_kopurua:	1	1	1
Eguna, gela, pertsona_kopurua:	1	5	1
.....			
Eguna, gela, pertsona_kopurua:	-4	-4	-4

2003/01/20-B KO AZTERKETAREN 6. ARIKETAREN EBAZPENA

```
#include <stdio.h>
#include <conio.h>

void Pantailaratu ();

int hotela [5][10]; //kanpo aldagaia

void main ()
{
    //erazagupena
    int ler, zut, datua, hutsa, partziala, guztiz, eguna, pertsona, kont;
    int kontmax, egunmax;

    clrscr();
    for (ler=0; ler < 5; ler++)
        for (zut = 0; zut <10; zut++)
            do {
                printf ("\n%d eguna, %d gela, zenbat pertsona?", ler, zut);
                scanf ("%d", &datua);
            } while (datua <0 || datua > 2);
            hotela [ler][zut] = datua;
        }
    Pantailaratu();
}

guztiz=0;
for (ler=0, hutsa=0, partziala=0; ler < 5; ler++)
    for (zut = 0; zut <10; zut++)
        if (hotela [ler][zut] == 0)
            {hutsa++;}
        else if (hotela [ler][zut] == 1)
            {partziala=partziala + 1;}
        else
            {guztiz++;}
    }
}

printf ("\nHutsik : %2d \nPartzialki : %2d \n Guztiz beteta: %2d\n", hutsa, partziala, guztiz);

printf ("Ze egun, ze pertsona kopurua : ");
scanf ("%d,%d", &eguna, &persona);

while (eguna<0 || eguna>14 || pertsona <= 0)
    {
        printf ("Datu txarrak. Ze egun, ze pertsona kopurua : ");
        scanf ("%d,%d", &eguna, &persona);
    }

for (zut = 19, kont = 0; zut>=0; zut--)
    if (hotela[eguna][zut] == 0)
        {kont = kont + 2;}
}
```

```

}
if (kont >= pertsona)
    {puts ("nSartzen dira");}
else
    {puts ("nEz dira sartzen");}
}

```

```

//////////d arketia
kontmax = -1;
egumax = -1;
for (ler=0, hutsa=0, partziala=0; ler < 5; ler++)
    {for (zuti = 0, kont = 0; zuti < 10; zuti++)
        {if (hotel[ler][zuti] == 0)
            {kont ++;}
        }
    }
}

```

```

printf ("%d egunean %d gela libre", ler, kont);
if (kontmax < kont)
    {kontmax = kont;
    egumax = ler;}
}

```

```

printf ("%d egunean gela libre gehien daukagu, %d", egumax, kontmax);
}

```

```

puts ("Sakatu tekla...");
getch();
}

```

```

void Pantalaratu ()
{int ler, zuti;

```

```

for (ler=0; ler < 5; ler++)
    {for (zuti = 0; zuti < 10; zuti++)
        {printf ("%2d ", hotel[ler][zuti]);
        }
    }
}

```

```

puts ("\n-----");
}
}

```



Industri Ingeniaritza Teknikoko
Unibertsitate Eskola
 Avda. Oteola, 29. 20600 ETIBAR. (Galditzcoa)
 943 03 3020 fax 943 03 3110
LENGOAI ETA SISTEMA INFORMATIKO saila



Informatikaren Oinarriak-en AZTERKETA.
1 IIT MEKANIKA TALDEA 31. 04-01-19. Ohiko delaldia.

IZEN ABIZENAK	NAN	NOTA
---------------	-----	------

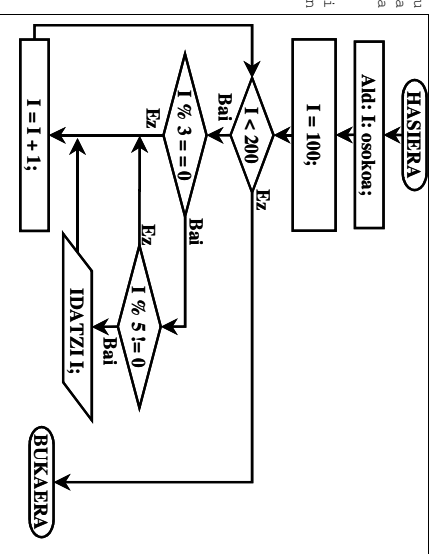
1: Puz-ren osagetak zeintzuk diren eta beraien eginbeharra zein den hitz gutxitan azaldu.

2: Ordenagailu batek 32 hariko helbide-busa eta 4 bytetako hitzak baditu, ze tamaina izango du RAM memoriak?

3: Ondorengo konbertsioak egin tarteko pausunak erakutsiz.
 (1.5) $(1011101,11010)_2 = (X)_8 = (Y)_{16} = (Z)_{10}$

4: Fluxu-diagrama hau jarraituz zer azalduko da pantailaz? Azalpena zehatza eta argia izan dadila.

Sortu Fluxu-diagramari dagokion C lengoaiari idatzitako programa.



5: Ondorengo egiten duen C programa bat sortu: erabiltzaileak emandako karaktere kate bat irakurri eta si taulan gorde; adierazi zehat bokal, kontsonante eta zuriuneak dauden si taulan.

6: Denda batetan 10 arrain mota saltzen dituzte, 5 arrain-ontzian gordetzen dituztelarik. Arrain-ontzi bakoitzean mota desberdineko arrainak nahas daitezke. Mena duen eta errepikakorria den C programa bat idatzi ondorengo egiten duena:

- DATUEN SARERRA: programak erabiltzaileari bahan-bahan arrain-ontzi bakoitzean arrain mota bakoitzeko ze kopuru dagoen eskatuko dio. (Arrain-ontzi X Mota Y Kopurua: Z).
- ARRAINEN SALMENTA: erabiltzaileak ze arrain mota, zein arrain-ontzitik eta ze kopuru saltzen den zehatzi ahal izango du. Programak datu hauek baretzatuko ditu salmenta onartu baino lehen. Arrain-ontzian mota horretako arrain nahiko egon behar du, bestela salmenta ez da egingo. Mota horretako nahiko arrain badago orduan zehaztutako kopurua deskontatuko da arrain-ontzitik.
- GAINPOPULAZIO DUEEN ARRAIN-ONTZIA: arrain-ontzi batek gainpopulazioa izango du baldin eta 15 arrain baino gehiago baditu. Programak arrain-ontzi batean zeharkia eskatuko du eta ea gainpopulazioa duen ala ez esango du.
- IZAKINAK: programak dendan dagoen mota bakoitzeko arrain kopurua adieraziko du.


```

clrscr();
printf("\n\n%c ARRAINONTZIEN DATU SARRETA: \n\n");
for (arrainontzia=0; arrainontzia < 5; arrainontzia++)
{
    for (arrainmota=0; arrainmota < 10; arrainmota++)
    {
        printf("\tArrainontzia %d Arraina %d KANTITATUA: ",
            arrainontzia+1, arrainmota+1);
        scanf("%g", &denda[arrainontzia][arrainmota]); /* DATUA
        while (denda[arrainontzia][arrainmota] < 0)
            printf("\t\terror. Idatzi berriro: ");
        scanf("%d", &denda[arrainontzia][arrainmota]);
    }
}

case '2':
clrscr();
printf("\n\n%c ARRAINEN SAIMENTA: \n\n");
scanf("%d", &arrainmota);
scanf("%d", &arrainmota);
while ((arrainmota < 1) || (arrainmota > 10)) /* DATUA BALIOZTATU
*/
{
    printf("\t\terror. Idatzi berriro: ");
    scanf("%d", &arrainmota);
}
printf("\tZenbat arrai?: ");
scanf("%d", &kantitatea);
while (kantitatea < 1)
/* DATUA BALIOZTATU */
{
    printf("\t\terror. Idatzi berriro: ");
    scanf("%d", &kantitatea);
}
printf("\tZenbat arrai?: ");
scanf("%d", &kantitatea);
while (kantitatea < 1) || (arrainontzia > 5))
/*
*/
{
    printf("\n\n\t%c %d. ARRAINONTZIAK EZ DITU %d ARRAI %d. MOTAKOK"
        , arrainontzia, arrainmota);
}
break;
}

case '3':
clrscr();
printf("\n\n%c GAINPOPULAZIOAREN ANALISIA: \n\n");
printf("\tZenbat arraiontzietatik: ");
scanf("%d", &arrainontzia);
while ((arrainontzia < 1) || (arrainontzia > 5))
/*
*/
{
    printf("\t\terror. Idatzi berriro: ");
    scanf("%d", &arrainontzia);
}
for (kantitatea=0, arrainmota=0; arrainmota < 10; arrainmota++)
{
    kantitatea += denda[arrainontzia-1][arrainmota];
}
}
}

```

```

if (kantitatea > 15)
{
    printf("\t\terror. arrain-ontziak GAINPOPULAZIOA du.\n",
        arrainontzia);
}
break;
}

case '4':
clrscr();
for (arrainmota=0; arrainmota<10; arrainmota++)
{
    arrainkopurua[arrainmota] = 0;
    for (arrainontzia=0; arrainontzia<5; arrainontzia++)
    {
        arrainkopurua[arrainmota] +=
            denda[arrainontzia][arrainmota];
    }
    printf("\n\n\t Dendan %d arrain daude %d. motakoak"
        , arrainkopurua[arrainmota],
        arrainmota+1);
}

case '5':
clrscr();
printf("\n\n%c ARRAINONTZIEN EDUKIA: \n\n");
for (arrainontzia=0; arrainontzia < 5; arrainontzia++)
{
    for (arrainmota=0; arrainmota < 10; arrainmota++)
    {
        printf("%7d", denda[arrainontzia][arrainmota]);
    }
    puts("");
}
break;
}

/* switch bukaera */
}

if (auk i=6)
{
    printf("\n\n\n\n\n\t Menua bueltatzeko sakatu tekla bat ... ");
    getch();
}
dejenon dei program */

clrscr();
puts("\n\n\t%c MENUA \n\n"); /* PROGRAMAREN MENUA */
puts("\t1 - Arrain-ontzien datu sarreta.");
puts("\t2 - Arrainen salmenta.");
puts("\t3 - Arrain-ontzi batean gainpopulazioa.");
puts("\t4 - Dendan izakinak.");
puts("\t5 - Arrain-ontzien edukia ikusi.");
puts("\t6 - PROGRAMATIK IRTEN.");
printf("\n\tSakatu zure aukera: ");
do
{
    auk = getch();
}
while (auk < '1' || auk > '6'); /* AUKERA TXARRAK EZEZTATU */
}

while (auk i=6); /* do-while bukaera */
return 0;
}

/* main bukaera */
}
}

```

1 IIT MEKANIKA TALDEA 31. 04-09-07. Ez-ohiko deialdia.

IZEN ABIZENAK	NAN	NOTA

- 1: a) Azaldu eta erlaziozatu kontzeptu hauek: **INFORMAZIOA**, **INTERPRETATZIOA** eta **DATUA**.
b) Zeintzuk dira Datu Prozesaketaren pasu nagusiak?
c) Ondorengo kontzeptuak definitu: **SINKROPIA**, **FLUXU-DIAGRAMA**, **PROGRAMA**.

- 2: 4 Terabyte dituen memoria bat 64 bit-eko hitzetan zatituta dago.
a) Memori honetako posizio guztiak arazitzeko zehab hari izan behariko ditu helbide bursak?
b) Eta hitzak 4 byte-koak balira?

- 3: Ondorengo konbertsioak egin tarteko pausuk erakutsiz:
(1) $(1D/Fl)_{16} = (X)_3 = (Y)_2 = (Z)_8$

- 4: **ESALDI BAKAR BATEAN** adierazi zain emaitza bueltatzen duen programa honet res aldagailan.

```
#include <stdio.h>
#include <conio.h>
int main()
{
    int x, y, res, i;
    printf("\n\Sartu lehen zerbakia: "); scanf("%d", &x);
    printf("\n\Sartu bigarren zerbakia: "); scanf("%d", &y);
    res = 0; i = 0;
    while (i < x)
    {
        res += y; i++;
    }
    printf("\n\Emaitza da: %d", res);
    return 0;
}
```

- 5: 0 zerbakiak bukatzen den osoko zerbaki ez-negatiboen errenkada bat irakurri eta era honetan prozesatzen dituen programa bat idatzi (0 irakurtzen denean bukatzen da errenkada):
• Zerbakia > 100 eta 3-ren anizkoltza bada: zati 3 egin eta emaitza pantailaz erakutsi.
• Zerbakia <= 100 eta 5-en anizkoltza, edo zerbakia > 100 eta 3-ren anizkoltza EZ bada: bider 2 egin eta emaitza pantailaz erakutsi.
• Beste zerbaki guztiak zuzenean pantailaz erakutsi.

- 6: Irakasle batek 100 ikasle matrikulatuta dituen irakaskalaren notak kudeatzeko programa bat nahi du. Ikaslearen Nota Finala hiru nota partzialen arabera kalkulatzeko da: azterketa, laborategi eta lanaren notak (%70 azterketa + %20 lana + %10 laborategia). Nota guztiak 0 eta 10 arteko zerbakiak dira eta horrela gordeko dira programan, nozarik ez balego -1 jarriko da. Programak ondorengoa egitea ahalbidetu du:

1. Azterketaren notak gorde. (Guztiak aldi berean eta banan-banan sartzen dira).
2. Laborategi-talde baten notak gorde. (20 ikasleen taldeka sartzen dira zerrendaren arabera: 1. taldea, 1-tik 20-ra; 2. taldea, 21-tik 40-ra, etab.).
3. Lanen notak gorde. (Ikasleek lanak entregatu ahala sartzen dira notak, banaka ala multzoka sar ditezke, beraz datu sarreara ikaslearen zerbakia 0 dela adierazten denean bukatuko da).
4. Ikasle baten Nota Finala kalkulatu. Ez daukan notak 0 puntu bezala hartuko dira.
5. Erabiltzaileak aukeratzeko duen laborategi-talde bateko ikasleen hiru nota partzialak eta Nota Finala erakutsi pantailaz.
6. Azterketa noten estatistikak egin: Ezazu notak, Aurkez notak, eta aurkez notakoen artean Ezazu notak eta Gaiztu notak. Adierazi multzo bakoitzeko guztira zehab ikasle eta balta ehuneko ere.

Informatikaren Oinarriak-en AZTERKETA.
1 IIT MEKANIKA TALDEA 31. 04-09-07. Ez-ohiko deialdia.

EBAZPENAK

2. arriketa:

D = C / P, 2ⁿ = D
C: memoriaren edukiera
P: hitzaren tamaina
N: helbide busaren hari kopurua

- 64 bit-eko hitzak:
4 Tbyte / 64 bit = 2^{2*10*2³} bit / 2⁶ bit = 2¹⁵/2⁶ = 2⁹ helbide => 39 hari
4 Byte-eko hitzak:
4 Tbyte / 4 Byte = 2^{2*10*2³} bit / 2^{2*2³} bit = 2¹⁵/2⁶ = 2⁹ helbide => 40 hari

3. arriketa:

Hamasetar	1	D	F	1
Bitar	0001	1101	1111	0001
Zortzitar	0 00	011	101	100
	0	3	5	4

$(1D, Fl)_{16} = 1*16^3 + 13*16^0 + 15*16^{-1} + 1*16^{-2} = 16 + 13 + 15/16 + 1/16^2 = (29,9414)_{10}$

$2 \quad 9 \quad \begin{matrix} | & 3 & \\ 9 & & \end{matrix} \quad \begin{matrix} | & 3 & \\ 3 & & \end{matrix} \quad \begin{matrix} | & 3 & \\ 3 & & \end{matrix} \quad \begin{matrix} | & 3 & \\ 1 & & \end{matrix}$

$0, \quad 9 \quad 4 \quad 1$
 $2, \quad 8 \quad 2 \quad 3$
 $2, \quad 4 \quad 6 \quad 9$
 $1, \quad 4 \quad 0 \quad 7$
 $1, \quad 2 \quad 2 \quad 1$

$(1D, Fl)_{16} = (1002, 2211)_3 = (11101, 111100010)_2 = (35, 742)_8$

4. arriketa:

Programa honek res aldagailan x eta y arteko biderkadura kalkulatzeko du.

x	y	Res	i	Pantaila
				Sartu ...
3				Sartu ...
	5			
	0		0	
	5		1	
	10		2	
	15		3	Emaitza da: 15
x <th>y</th> <th>res</th> <th>i</th> <th>Pantaila</th>	y	res	i	Pantaila
				Sartu ...
4				Sartu ...
	2		0	
	0		0	
	4		1	
	8		2	Emaitza da: 8


```

ikaslea+1);

printf("\t %d. ikaslearen azterketa nota sartu: ",
scanf("%f", &infor[0][ikaslea]));
/* datua baieztatu */
while ( (infor[0][ikaslea]<-1)
|| ((infor[0][ikaslea]>-1)&&(infor[0][ikaslea]<0))
|| (infor[0][ikaslea]>10) )
{
    printf("\t\toker. Sakatu berriro: ");
    scanf("%f", &infor[0][ikaslea]);
}
break;

case 'B':
/* Laborategi-talde baten notak sartu */
clrscr();
printf("\n\tlaborategi-talde baten notak sartu ... \n\n");
printf("\t\taldeen zehakia zehaztu (1 - 5): ");
scanf("%d", &taldea);
while (taldea < 1 || taldea > 5) /* datua baieztatu */
{
    printf("\t\toker. Sakatu berriro: ");
    scanf("%d", &taldea);
}
switch (taldea)
{
    case 1: ikaslea = 0; break;
    case 2: ikaslea = 20; break;
    case 3: ikaslea = 40; break;
    case 4: ikaslea = 60; break;
    case 5: ikaslea = 80; break;
}

for (f = ikaslea; f < ikaslea + 20; f++)
{
    printf("\t %d. ikaslearen laborategiko nota sartu: ", f+1);
    scanf("%f", &infor[1][f]);
    /* datua baieztatu */
    while ( (infor[1][f]<-1)
|| ((infor[1][f]>-1)&&(infor[1][f]<0))
|| (infor[1][f]>10))
    {
        printf("\t\toker. Sakatu berriro: ");
        scanf("%f", &infor[1][f]);
    }
    break;
}

case 'C':
/* Lanen notak sartu */
clrscr();
printf("\n\tlanen notak sartu ... \n\n");
for ( ; 1; )
/*amaiera gabeko for*/
{
    printf("\n\tikasle zehakia zehaztu (0 - 100): ");
    scanf("%d", &ikaslea);
    while (ikaslea<0 || ikaslea>100) /* datua baieztatu */
    {
        printf("\t\toker. Sakatu berriro: ");
        scanf("%d", &ikaslea);
    }
    if (ikaslea == 0)
    {
        break;
    }
}

```

```

printf("\t %d. ikaslearen lanaren nota sartu: ", ikaslea);
scanf("%f", &infor[2][ikaslea-1]);
/* datua baieztatu */
while ( (infor[2][ikaslea-1]<-1)
|| ((infor[2][ikaslea-1]>-1) && (infor[2][ikaslea-1]<0))
|| (infor[2][ikaslea-1]>10))
{
    printf("\t\toker. Sakatu berriro: ");
    scanf("%f", &infor[2][ikaslea-1]);
}
break;

case 'D':
/* Ikasle baten
Nota Finala kalkulatu */
clrscr();
printf("\n\tikasle baten zehakia zehaztu (1 - 100): ");
scanf("%d", &ikaslea);
while (ikaslea<1 || ikaslea>100) /* datua baieztatu */
{
    printf("\t\toker. Sakatu berriro: ");
    scanf("%d", &ikaslea);
}
if (infor[0][ikaslea-1] == -1)
{
    infor[3][ikaslea-1] = 0;
}
else
{
    infor[3][ikaslea-1] = 0.7 * infor[0][ikaslea-1];
}
if (infor[1][ikaslea-1] == -1)
{
    infor[3][ikaslea-1] += 0;
}
else
{
    infor[3][ikaslea-1] += 0.2 * infor[1][ikaslea-1];
}
break;

case 'E':
/* Laborategi-talde baten nota partzialak
eta finala ikusi */
clrscr();
printf("\n\tlaborategi-talde baten notak... \n\n");
printf("\t\talde baten zehakia zehaztu (1 - 5): ");
scanf("%d", &taldea);
while (taldea<1 || taldea>5) /* datua baieztatu */
{
    printf("\t\toker. Sakatu berriro: ");
    scanf("%d", &taldea);
}
switch (taldea)
/* indizea hasieratu */
{
    case 1: ikaslea = 0; break;
    case 2: ikaslea = 20; break;
}
}

```

```

        case 3: ikaslea = 40; break;
        case 4: ikaslea = 60; break;
        case 5: ikaslea = 80; break;
    }
    printf("\n\n\t %d, taldearen notak: \n\n", taldea);
    for (F = ikaslea; F < ikaslea + 20; F++)
    {
        for (nota=0; nota<4; nota++)
        {
            printf("\tIkaslea %d: %10.2f", ikaslea+1,
                nota);
        }
        puts("");
    }
    break;
}

/* Azterketaren estatistikak */
clrscr();
printf("\n\tAzterketaren notak aztertuz ... \n");
for (EzAurkeztu=Aurkeztu=EzGai=0, ikaslea=0; ikaslea<100;
    ikaslea++)
{
    if (infor[0][ikaslea] == -1)
    {
        EzAurkeztu ++;
    }
    else
    {
        Aurkeztu++;
        if (infor[0][ikaslea] < 5)
        {
            EzGai++;
        }
        else
        {
            Gai++;
        }
    }
}

printf("\n\n\tEz Aurkeztutakoak: %d (ikasle guztien %% %d).",
    EzAurkeztu, (EzAurkeztu*100.0/(Aurkeztu+EzAurkeztu));
printf("\n\tAurkeztutakoak: %d (ikasle guztien %% %d).", Aurkeztu,
    (Aurkeztu*100.0/(Aurkeztu+EzAurkeztu));
if (Aurkeztu != 0)
{
    printf("\n\t\tEz Gaiditutakoak: %d (%0.2f %% de los
        Aurkeztu.), EzGai, (EzGai*100.0/Aurkeztu));
    printf("\n\t\tGaiditutakoak: %d (%0.2f %% de los
        Aurkeztu.), Gai, (Gai*100.0/Aurkeztu));
}
break;
}

/* switch */
printf("\n\n\tMenua bueltatzeko sakatu tekla bat ... ");
getch();
/* exekuzioa geldirik tekla bat sakatu arte */
while (1)
{
    return 0;
}

/* main() */
}

```