

INFORMATIKAREN OINARRIAREN AZTERKETA

2015eko ekainaren 15a

Izen-abizenak:..... Taldea.....

1. ARIKETA: INDIZE PILATZAILE-SEGIDA (3 puntu)

Ariketa egiterakoan, lehenengo eta behin, erabiltzaileari teklatutik *ElemKop* zenbaki bat sartzeko eskatuko zaio (*ElemKop* zenbakia 1etik 100era artekoa izango da). *ElemKop* sartu ostean, erabiltzaileari teklatutik bi aldiz, *ElemKop* zenbaki oso sartzeko eskatuko zaizkio, eta sartutako zenbakiak bi tauletan gordeko dira (A eta B tauletan).

Adibidez, erabiltzaileak *ElemKop* berdin 10 sartuz gero, A eta B taulak hurrengo eran betetzen dituela suposatuko dugu:

Erabiltzaileak A taula betetzeko teklatutik hurrengo zenbaki oso positiboak (≥ 0) sartu dituela suposatuko dugu:

A	3	7	1	12	45	23	4	56	2	20
	0	1	2	3	4	5	6	7	8	9

Ondoren, erabiltzaileak B taula betetzeko teklatutik hurrengo zenbaki indizeak ($[0, ElemKop - 1]$) sartu dituela suposatuko dugu (jakinda taulan sartutako indizeak ezin direla errepikatu):

B	3	7	1	2	5	8	4	6	0	9
	0	1	2	3	4	5	6	7	8	9

Programaren helburua A eta B taulako balioekin, hirugarren taula bat sortzea izango da (C taula izenekoa). C taulako elementuen balioak, hurrengo eran osatuko dira:

$$C[0]=0+A[B[0]]$$

$$C[1]=C[0]+A[B[1]]$$

$$C[2]=C[0]+C[1]+A[B[2]]$$

$$C[3]=C[0]+C[1]+C[2]+A[B[3]]$$

...

Aurreko adibidearen arabera, sortutako C taula, era honetan geratuko zen:

C	12	68	87	168	358	695	1433	2825	5649	11315
	0	1	2	3	4	5	6	7	8	9

Goian azaldutako programaren kodigoa idaztea eskatzen da. Horretarako ondoren aipatzen diren funtzioak definitu eta erabili:

a) **datuen_taula_eskatu:** Funtzioak *ElemKop* zenbaki bat eta zenbaki osoen taula bat emanda, erabiltzaileari *ElemKop* zenbaki eskatu eta taulan gordeko ditu. Funtzioak erabiltzaileak sartutako zenbaki guztiak osoak eta positiboak (≥ 0) direla egiaztatu behar du. Erabiltzaileak emandako zenbakia zuzena ez bada, funtzioak zenbakiak eskatzen jarraituko du ≥ 0 den zenbaki bat jaso arte, guztira *ElemKop* zenbaki oso positibo sartu arte. (0,3 puntu)

b) **(Zure azterketa notaren %100a baldin bada soilik) egiaztatu_taula** funtzioak *ElemKop* zenbaki bat eta zenbaki osoen taula bat emanda, funtzioak sartutako taularen elementuak $[0, ElemKop-1]$ balioen artean daudela eta ez direla errepikatzen egiaztatuko du. Taularen balioak baldintza betez gero, funtzioak 1 edo bestela 0 itzuliko du. (0,7 puntu)

- c) **eraiki_taula** funtzioak *ElemKop* zenbaki bat eta hiru taula (A, B eta C) emanda, A eta B taulen balioak erabiliko ditu, goiko definizioaren arabera C taula eraikitzeko. (1 puntu)
- d) **taula_ikusi** funtzioak *ElemKop* zenbaki bat eta taula bat emanda, funtzioak taularen edukina pantailatik aterako du. (0,25 puntu)
- e) **Programa nagusiak** lehenengo *Elemkop* zenbakia eskatuko dio erabiltzaileari, ostean goiko funtzioak erabiliz, A eta B taulak betetzeko eskatuko dizkio. Ondoren, programak C taula eraiki eta C taularen balioak pantailatik aterako ditu. Programa nagusiak **egiaztatu_taula** erabiliko du, B taula era egokian bete dela egiaztatzeko. B taula EZ egokia izanez gero, behin eta berriro B taularen balioak betetzeko aginduko dizkio erabiltzaileari. Denok erabili behar duzue **egiaztatu_taula** funtzioa, baina soilik inplementatu **zuen azterketa notaren %100a baldin bada**. (0,75 puntu)

2. ARIKETA: HITZEN TRANSFORMAZIOA (3 puntu)

Hitzen transformazioa eragiketa arrunt bat da. Oraingo honetan, programak erabiltzaileari behin eta berriz hitz bat sartzeko eskatuko dio, eta programak sartutako hitza transformatu ostean, transformatutako hitza pantailatik aterako du. OHARRA: erabiltzaileak hitz guztiak beti letra xehez sartzen dituela suposatuko dugu, hau da, ez da hitza letra xehez sartu dela baieztatu behar.

4 motako transformazioak egiteko, ondorengo arauak erabiliko dira:

1. Hitzaren karaktere kopurua bikoitia bada eta bokal kopurua bakoitia bada, hitz berria berdina baina atzekoz aurrera irakurrita izango da. Eta hitz berriaren hizki guztiak xeheak izango dira. (*examen* → *nemaxe*)
2. Hitzaren karaktere kopurua bakoitia bada eta bokal kopurua bikoitia bada, hitz berriaren lehenengo eta azkenengo hizki xeheak hizki larrietan transformatuko dira, eta gainerako hizki xeheak bere horretan mantenduko dira. (*hotel* → *HoteL*)
3. Hitzaren karaktere kopurua eta bokal kopurua bakoitia badira, hitz berriaren lehenengo eta azkenengo hizkiak xeheak izango dira eta gainerakoak hizki larrietan transformatuko dira. (*piscina* → *pISCINa*)
4. Hitzaren karaktere kopurua eta bokal kopurua bikoitiak badira, hitz berriaren lehenengo eta azkenengo hizkiak trukatu dira. Eta hitz berriaren hizki guztiak xeheak izango dira. (*casa* → *aasc*).

Goian azaldutako programaren kodigoa idaztea eskatzen da. Horretarako ondoren aipatzen diren funtzioak definitu eta erabili:

- a) **bokal_kopurua** funtzioak minuskulaz idatzitako hitz bat daukan kate bat emanda, funtzioak hitz horrek duen bokal kopurua itzuliko du. (0,25 puntu)
- b) **hitz_luzeera** funtzioak minuskulaz idatzitako hitz bat daukan kate bat emanda, funtzioak hitz horrek duen hizki kopurua itzuliko du. (0,25 puntu)
- c) **atzekoz_aurrera** funtzioak minuskulaz idatzitako hitz bat daukan kate bat emanda, sartutako hitza atzekoz aurrera idatzitako kate berri baten itzuliko du. (0,6 puntu)
- d) **lehen_azken_larritz** funtzioak minuskulaz idatzitako hitz bat daukan kate bat emanda, sartutako hitzaren lehenengo eta azkenengo hizki xeheak hizki larrietan transformatu eta gainerako hizki xeheak bere horretan mantendu ostean sartutako hitz berria kate berri baten itzuliko du. (0,3 puntu)
- e) **erdikoak_larritz** funtzioak minuskulaz idatzitako hitz bat daukan kate bat emanda, sartutako hitzaren lehenengo eta azkenengo hizkiak xehe bezala utziz eta gainerakoak hizkiak hizki larrietan transformatu ostean sartutako hitz berria kate berri baten itzuliko du. (0,3 puntu)

f) **lehen_azken_trukaketa** funtzioak minuskulaz idatzitako hitz bat daukan kate bat emanda, sartutako hitzaren lehenengo eta azkenengo hizkiak trukatu ostean sortutako hitz berria kate berri baten itzuliko du. (0,4 puntu)

g) **Programa nagusiak** etengabe erabiltzaileari edozein tamainako hitzak eskatuko dizkio (sartutako hitz bakoitzak gehienez 80 karaktere izango ditu). Aurreko ataletan azaldutako funtzioei deituta, sartutako hitz bakoitzari dagokion transformazioa burutu ostean, transformatutako hitzak pantailan agertuko dira. Prozesua **erabiltzaileak kate hutsa sartu arte errepikatuko da**. (0,6 puntu)

h) (**Zure azterketa notaren %100a baldin bada soilik**) **hitzaren_mota_kopurua** funtzioak esaldiko hitz guztietatik, zenbat hitz transformatu diren mota bakoitzeko (1, 2, 3 eta 4) pantailan agertuko da. (ikus beheko adibidea). (0,3 puntu)

Exekuzio-adibidea (erabiltzaileak teklatutako datuak **letra lodiz, etzanez eta azpimarratuta** ageri dira):

Sartu hitz bat letra xehez (Bukatzeko sakatu Intro hitz bezala): <i>el</i> Transformazioa: le	
Sartu hitz bat letra xehez (Bukatzeko sakatu Intro hitz bezala): <i>examen</i> Transformazioa: nemaxe	
Sartu hitz bat letra xehez (Bukatzeko sakatu Intro hitz bezala): <i>fue</i> Transformazioa: FuE	
Sartu hitz bat letra xehez (Bukatzeko sakatu Intro hitz bezala): <i>facil</i> Transformazioa: FaciL	
Sartu hitz bat letra xehez (Bukatzeko sakatu Intro hitz bezala): <i>Intro sakatu du</i> Mota bakoitzeko hitz kopurua:	
1 motako 2 hitz	¡ADI! %100eko azterketa egiten dutenenek honakoa lortu beharko lukete:
2 motako 2 hitz	
3 motako 0 hitz	
4 motako 0 hitz	

Oharra:

- **Guk proposatutako funtzioak erabili behar dira, horrez gain, nahi dituzunak programa eta erabil ditzakezu.**
- **string.h eta ctype.h goiburuko fitxategietan dauden funtzioak erabil ditzakezu.**

3. ARIKETA: ASTEKO DIRU-BILKETA (3 puntu)

Ordenagailuko katalogo batean, supermerkatu batean, gertatu diren asteko salmenten erregistroak daukaten fitxategiak aurki genezake. Salmenta-erregistro bakoitzak lau balio izango ditu: bezeroaren NAN zenbakia, erositako produktuaren *unitate* prezioa, unitate kopurua eta izena. Fitxategien izenak, asteko egunen izenak izango dira, eta bakoitzean egun horretan erositako erregistroak egongo dira, adibidez, "astelehena.txt", "asteartea.txt", ..., "larunbata.txt". Beheko adibidean fitxategi bakoitzaren erregistroak lerroz-lerro agertzen dira:

Adibidez: "astelehena.txt" fitxategiaren edukina:

```
12345678-A 2.56 2 NUTELLA
12345678-A 1 3 OGIA
45123476-H 3 6 LARANJA
```

"asteartea.txt" fitxategiaren edukina:

```
82537486-J 1.5 1 ESNEA
12345678-A 1 3 OGIA
```

Supermerkatuan asteko 6 egunetan (astelehenetik larunbatera) egindako salmentak "kutxa.txt" fitxategian erregistratzen dituen kodigoa idaztea eskatzen da. Erregistratuko den informazioa hurrengoa izango da: pertsona batek guztira egunean zehar egindako erosketak eta guztira egun horretan bildutako diru-kopurua, eta fitxategiaren azken lerroan astean zehar supermerkatuan bildutako guztia idatziko da.

Adibidez: Adibideko goiko datuekin sortutako "kutxa.txt" fitxategiaren edukina:

```
1. EGUNEAN BILDUTAKO DIRU-KOPURUA:
12345678-A guztira: 8.12
45123476-H guztira: 18
    GUZTIRA: 26.12
2. EGUNEAN BILDUTAKO DIRU-KOPURUA:
82537486-J guztira: 1.5
12345678-A guztira: 3
    GUZTIRA: 4.5
3. EGUNEAN BILDUTAKO DIRU-KOPURUA: ...:
...
...
6.      EGUNEAN BILDUTAKO DIRU-KOPURUA: ...
...
GUZTIRA BILDUTAKO DIRU-KOPURUA: ....
```

Goian azaldutako programaren kodigoa idaztea eskatzen da. Horretarako ondoren aipatzen diren funtzioak definitu eta erabili:

a) fitxategia_ireki: Funtzio honi 1 eta 6 arteko zenbaki oso bat emanda, egun horretako fitxategia irakurtzeko soilik zabaldu eta irekitako fitxategiaren erakuslea itzultzen du. Adibidez, 1 zenbakia sartuz gero "astelehena.txt" fitxategia zabalduko du, 2 zenbakia sartuz gero "asteartea.txt" fitxategia, ..., eta 6 zenbakia sartuz gero "larunbata.txt" fitxategia zabalduko du, eta irekitako fitxategiaren erakuslea (**FILE*) itzuliko du. Funtzioaren burua horrela geratuko zen: *FILE *fitxategia_ireki(int eguna)*. (0,5 puntu)

b) nan_berdina: funtzioak NAN bi kate emanda, funtzioak pasatutako NAN biak berdinak izanez gero 0 bat itzuliko du, bestela 1 itzuliko du. (0,25 puntu)

c) eguneko_fitxategia_tratatu: funtzio honen burua hurrengoa izango da: float eguneko_fitxategia_tratatu(FILE * sarrera, FILE * irteera). Funtzioak sarrera fitxategitik eguneko salmentak irakurriko ditu, eta pertsona batek guztira egunean zehar egindako erosketak eta guztira egun horretan bildutako diru-kopurua irteerako fitxategian erregistratuko ditu. Funtzioak guztira egun horretan bildutako diru-kopurua itzuliko du. (1,5 puntu)

d) Programa nagusiak: aurreko ataletan azaldutako funtzioei deituta, asteko 6 eguneko fitxategien informazioa irakurri ostean, egindako salmentak "kutxa.txt" fitxategian erregistratuko ditu. "kutxa.txt" fitxategian erregistratuko den informazioa hurrengoa izango da: pertsona batek guztira egunean zehar egindako erosketak eta guztira egun horretan bildutako diru-kopurua, eta fitxategiaren azken lerroan astean zehar supermerkatuan bildutako guztia idatziko da. (0,75 puntu)