



## INFORMATIKAREN OINARRIAK - AZTERKETA

### 2013-ko Ekainak 27a

#### 1. ARIKETA: METROAREN KUDEAKETA (3 puntu)

Metro Bilbao enpresarentzat programa bat egin nahi dugu. Programa honek bidaiariak linea gorriko (Plentzia-Etxebarri) zein geltoki erabiltzen duten gehien egiaztatzen du. Hau da, dauden 28 geltokietatik, zeintzuen erabilera dagoen batez bestekoaren gainetik.

Hurrengo hartu behar da kontutan:

- Metro-linea bi noranzkoa da. Beraz, bidaiariak noranzko bietan joan daitezke.
- Metro-linea ez da zirkularra. Beraz, Plentzia-Etxebarri noranzkoan, azkenengo geltoki posiblea 28-a da. Eta Etxebarri-Plentzia noranzkoan, azkenengo geltoki posiblea 1-a da.
- 1 geltokia Plentzia da eta 28 geltokia Etxebarri da.

28 posizio (0-tik 27-ra) dituen array bat erabiliko da geltoki bakoitza zenbat bidaiarik erabili duten jakiteko. Posizio guztiak zerora hasieratu beharko dira.

Geltokiak

0	0	0	0	0	....	0
0	1	2	3	4	....	27

Bidai bakoitzeko, array-ak ekintza guztiak erregistratu egingo ditu (igo, jarraitu, jaitsi).

Adibidez

Erabiltzaile bar 3. Geltokian igotzen bada eta 7 geltokian jaisten bada, hurrengo idazpena gertatuko da:

Geltokiak

0	0	1	1	1	1	1	0	....	0
0	1	2	3	4	5	6	7	....	27

Hurrengo erabiltzailea 6. Geltokian igotzen bada eta 8. Geltokian jaisten bada, hurrengo idazpena gertatuko da:

Geltokiak

0	0	1	1	1	2	2	1	....	0
0	1	2	3	4	5	6	7	....	27

Metro Bilbao-ri batez besteko erabileraren gainetik dauden geltokian kalkulatzeko laguntzen duen programa burutzea eskatzen da. Horretarako hurrengo definitu behar da:

- a) **datuak\_irakurri** funtzioa: erabiltzaileari igoera-geltoki bat eta jaitsiera-geltoki bat eskatzen dizkio (komaz bananduta) eta geltoki-zenbakiak egokiak diren egiaztatzen du (kontutan hartu metro-linearen berezitasunak:bi zentzuko eta ez zirkularra). Baiezko kasuan, *geltokiak\_eguneratu* funtzioa deituko da, geltokien array-a eguneratzeko. Kontrako kasuan, erabiltzailea abisatuko du pantailan mezu bat erakutsiz eta datak berriro eskatuko ira zuzenak izan arte (**0,75 puntu**).



- b) **geltokiak\_eguneratu** funtzioa: bi geltoki zenbaki eta geltokien array-a emanda, pasatzen den geltoki guztien posizio guztiak inkrementatzen ditu (igoerakoa eta jaitsierakoa barne) (*puntu 0,75*).
- c) **batazbestekoa\_kalkulatu** funtzioa: zenbaki osoen array baten batez bestekoa itzultzen du (*0,25 puntu*).
- d) **geltokiak\_erakutsi** funtzioa: geltokien array-a emanda eta *batazbestekoa\_kalkulatu* funtzioa erabiliz, geltokien batez besteko erabilera lortzen du eta balio hau eta honen gainera erabiltzen diren geltokiak pantailaratzen ditu (*0,5 puntu*).
- e) **Programa nagusia**: aurreko ataletan deskribatutako funtzioen erabilera zuzena edo indirektoa eginez, geltokien arraya 0-ra hasieratzen du. Gero erabiltzaileari geltokiak eskatzeko prozesua hasten da, honek nahi duen bitartean. Geltokien datu sarrera amaitzen denean, geltokien batez besteko erabilera eta horren gainera erabiltzen diren geltokiak pantailaratzen ditu (*0,75 puntu*).

Exekuzio adibidea (teklatu bitartez sartutako datuak letra etzan, lodi eta azpimarratuan agertzen dira):

Sartu igoera eta jaitsiera geltokiak (komaz bereizita): 0, 14

Sartutako datuak ez dira zuzenak.

Sartu igoera eta jaitsiera geltokiak (komaz bereizita): 18, 3

Datu gehiago sartu nahi duzu? (b/e): b

Sartu igoera eta jaitsiera geltokiak (komaz bereizita): 2, 25

Datu gehiago sartu nahi duzu? (b/e): n

Geltokien batez besteko erabilera hau da: 1,43

Batez bestekoaren gainera erabiltzen diren geltokiak hauek dira: 3, 4, 5, 6, 7, 8, 9 10, 11, 12, 13 14, 15, 16, 17, 18,

Sakatu tekla bat jarraitzeko.....



## 2. ARIKETA: KATE BATEN KODIFIKAZIOA (3 puntu)

Letra maiuskulaz eta zurienez bakarrik osaturiko esaldi baten oinarrituz kate bat kodifikatu eta pantailaratu nahi da. Kate kodifikatua letra maiuskulen ordezkari minuskulak jarri, 'A' bakoitzaren ordezkari '3' jarri, zuriune guztiaz ezabatuz, hitzak banatzeko '7' bat jarri eta geratzen den esaldiari buelta emanaz lortzen da.

Programa nagusiak ondorengo atazak buruko ditu orden honetan (eskatzen diren baino funtzio gehiago definitzeko aukera dago):

1. Gehienez 250 karaktere duen kate bat **eskatu**, letra maiuskulaz eta zurienez soilik osatuta dagoela **egiaztatuz (0,25 puntu)**. Egiaztapen hau burutzeko, derrigorrez hurrengo funtzioak definitu eta erabili behar izango dira:
  - a) **maiuskula\_edo\_zuriunea\_da**, karaktere bat emanda, 1 itzultzen du maiuskula edo zuriunea denean eta 0 ez denean honetariko bat. **(0,25 puntu)**
  - b) **kate\_egokia\_da**, kate bat emanda 1 itzultzen du letra maiuskulaz eta zurienez soilik osatuta badago, eta 0 itzultzen du ezezko kasuan. **(0,25 puntu)**

i: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 ... 250

'L'	'A'	'	'	'V'	'A'	'C'	'A'	'	'D'	'I'	'J'	'O'	'	'M'	'U'	'	'\0'		
-----	-----	---	---	-----	-----	-----	-----	---	-----	-----	-----	-----	---	-----	-----	---	------	--	--

2. **Kate berri bat sortu** hurrengo aldatetekin: 'A' bakoitzaren ordezkari '3' jarri, gainerako letra guztiak minuskuletara pasatu <sup>1</sup>, zuriune guztiak ezabatuz eta '7' hitzen artean tartekatuz (zuriune bat baino gehiago dagoenean, '7' bakarra jarriko da, adibidean erakusten den moduan). Kontutan hartu eskatutako katean zenbait zuriune agertu daitezkeela, bai kate hasieran, bai amaieran. Hauek ez dute agertu behar transformaturako katean. **(1,25 puntu)**

j: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 ... 250

'l'	'3'	'7'	'v'	'3'	'c'	'3'	'7'	'd'	'i'	'j'	'o'	'7'	'm'	'u'	'\0'				
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	------	--	--	--	--

3. Bigarren puntuan sortutako kateari **buelta eman. (0,75 puntu)**

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 ... 250

'u'	'm'	'7'	'o'	'j'	'i'	'd'	'7'	'3'	'c'	'3'	'v'	'7'	'3'	'l'	'\0'				
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	------	--	--	--	--

4. Hirugarren pausuan sortutako katea **bistaratu. (0,25 puntu)**

<sup>1</sup> Letra maiuskula bat minuskula bihurtzeko ondorengo adierazpena erabili daiteke: *letramaius* + 'a' - 'A'. Bestela, *ctype.h* librerian definitzen den char **tolower** (char *kar*) funtzioa erabili daiteke, *kar* karakterea minuskula bihurtzeko.



Exekuzio adibidea (teklatu bitartez sartutako datuak letra etzan, lodi eta azpimarratuan agertzen dira):

Esaldi bat sartu (gehienez 250 karaktere): 13 egunean 20:30-tan

ERROREA. Esaldia letra maiuskulaz eta zurinez soilik osatuta egon behar da.

Esaldi bat sartu (gehienez 250 karaktere): LA VACA DIJO MU

Esaldi kodifikatua: um7ojid73c3v73l

*Sakatu tekla bat amaitzeko...*



### 3. ARIKETA: FITXATEGIAK (3 puntu)

Aparkaleku baten arduradunek okupazioa kudeatzen laguntzen dien programa bat nahi dute, aparkalekuan sartzen diren erabiltzaileak solairu bakoitzean libre dauden lekuei buruz informatzeko. Aparkalekuak 7 solairu eta 100 leku ditu solairu bakoitzeko ditu.

Erabiltzaile bat sartzen den momentuan, programa nagusiak aparkalekuaren egoera fitxategi batetik irakurri behar du, 7 posizioko array baten gorde eta, arraya-ren informaziotik aurrera, beste fitxategi baten solairu bakoitzean libre geratzen diren lekuen laburpena idatzi.

Sarrera eta irteera fitxategien formatua b eta c ataletan azaltzen da. Fitxategi hauen izenak aparkalekuaren erabiltzaileari eskatuko zaizkio programaren exekuzioan zehar.

Programa nagusia (0,75 puntu) eta zuzenean edo indirektoki programa nagusian erabiltzeko ondoren deskribatzen diren funtzioak inplementatu:

- a) **fitxategi\_izena\_eskatu:** erabiltzaileari fitxategi izen bat eskatzen dio (karaktere katea) eta itzultzen du. (0,25 puntu)
- b) **hasierako\_datuak\_kargatu:** funtzio honek egungo aparkalekuaren egoera gordetzen duen fitxategiaren izena eta 7 posizio dituen array bat jasotzen ditu. Funtzioak fitxategitik aparkalekuaren okupazio datuak irakurri eta array-an solairu bakoitzean libre dauden lekuen kopurua gordetzen du. Fitxategiak 7 lerro ditu (solairu bakoitzeko, bana) eta lerro bakoitzean komaz banandutako bi zenbaki oso daude, lehenengoak okupatutako lekuak adierazten ditu eta bigarrenak okupatu ez daitezkeen lekuak (erreserbatuta daudelako edo erabili ezin direlako). Solairu bakoitzean dauden leku libreen kopurua honela kalkulaten da: leku totalak (100) ken okupatutako lekuak, ken erreserbatutako lekuak. Adibidez, bigarren solairuan: 75 libre = 100 totalak – 23 okupatuta – 2 erreserbatuta. (puntu 1)

Sarrera fitxategi baten adibidea:

79,0
23,2
43,3
2,0
...

- c) **solairu\_bakoitzeko\_leku\_libreak\_idatzi:** funtzio honek solairu bakoitzean libre dauden lekuen array-a eta erabiltzailea informatzeko erabiliko den fitxategiaren izena jasotzen ditu. Funtzioak fitxategi horretan solairu bakoitzean libre dauden lekuei buruzko informazioa gorde behar du, adibidean erakusten den formatua erabiliz. (puntu 1)

Irteera fitxategiaren adibidea:

#### APARKALEKUKO EGUNGO EGOERA

- 1. Solairua: 21 leku libre
- 2. Solairua: 75 leku libre
- 3. Solairua: 54 leku libre
- 4. Solairua: 98 leku libre
- ...



Exekuzio adibidea (teklatu bitartez sartutako datuak letra etzan, lodi eta azpimarratuan agertzen dira):

1. Pausua: aparkalekuko egungo egoera kargatu

Sartu sarrera fitxategiaren izena:

Egoera11-53.txt

Datuak kargatu dira.

2. Pausua: Fitxategia idatzi aparkalekuaren erabiltzailea informatzeko.

Sartu irteera fitxategiaren izena:

InformazioaErabiltzailearentzat.txt

Fitxategia ondo sortu da.

Sakatu tekla bat amaitzeko...