



INFORMATIKAREN OINARRIAK – AZTERKETA
2014ko uztailaren 3a

EJERCICIO 1: BEKTOREEN ARTEKO EBAKIDURA (3 puntu)

A eta B bektoreen ebakidura aldi berean A-renak eta B-renak diren elementuek osatzen duten C bektorea da. Adibidez:

A bektorea:

30	71	22	-5	?	?
0	1	2	3	4	99

B bektorea:

71	-2	30	4	-5	?	?
0	1	2	3	4	5	99

C bektorea = A eta B-ren ebakidura

30	71	-5	?	?
0	1	2	3	99

Erabiltzaileari bi bektore eskatu eta beraien ebakidura ateratzen duen programa bat inplementatu. Hiru bektoreek tamaina berekoak izan behar dute (nahiz eta erabilitako posizio kopuruak desberdinak izan) eta ezin dute osagai errepikaturik izan. Horretarako, hurrengo funtzioak inplementatu eta erabili behar dira:

- a) **bektorea_irakurri**: bektore bat emanda, zenbaki osoak eskatzen dizkio erabiltzaileari eta bektorean sartzen ditu. Irakurketa bukatzen da baldin eta bektorea betetzen bada edo erabiltzaileak 0 idazten badu. Funtzioak bektorean sartutako osagai kopurua bueltatzen du. **(0.25 puntu)**
- b) **barne_dago**: x zenbaki osoa, B bektorea, eta B-ren osagai kopurua emanda, ea x B bektorean agertzen den edo ez erabakitzen du. Baiezkoan funtzioak 1 bueltatzen du; bestela 0. **(0.5 puntu)**
- c) **ebakidura**: A, B eta C bektoreak, eta A eta B osagai kopuruak emanda, B bektorean agertzen diren A-ren osagaiak C-en sartzen ditu. C bektorean sartutako osagai kopurua bueltatzen du. **(puntu 1)**
- d) **bektorea_idatzi**: bektore bat eta bere osagai kopurua emanda, bektorearen osagaiak pantailaratzen ditu. **(0,25 puntu)**
- e) **programa nagusia**: aurreko funtzioak zuzen edo zeharka erabiliz, bi bektore eskatzen dizkio erabiltzaileari eta beraien ebakidura pantailaratzen du. Ondoren, erabiltzaileari galdetzen dio ea datu gehiago sartu nahi duen edo ez. Erantzuna baiezkoa bada, prozesu osoa errepikatzen da, bestela programa bukatzen da. Programak konprobatu behar du erabiltzaileak 'b' edo 'e' erantzuten duela, bestela berriro galdetu behar dio. **(puntu 1)**



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

Lehenengo bektorea:
Bektorerako elementu bat sartu (zero amaitzeko): 30
Bektorerako elementu bat sartu (zero amaitzeko): 71
Bektorerako elementu bat sartu (zero amaitzeko): 22
Bektorerako elementu bat sartu (zero amaitzeko): -5
Bektorerako elementu bat sartu (zero amaitzeko): 0
Bigarren bektorea:
Bektorerako elementu bat sartu (zero amaitzeko): 71
Bektorerako elementu bat sartu (zero amaitzeko): -2
Bektorerako elementu bat sartu (zero amaitzeko): 30
Bektorerako elementu bat sartu (zero amaitzeko): 4
Bektorerako elementu bat sartu (zero amaitzeko): -5
Bektorerako elementu bat sartu (zero amaitzeko): 0

Ebakidura: 30 71 -5

Berriro hasi nahi duzu? (b/e): a
Errorea: b edo e sartu behar duzu.
Berriro hasi nahi duzu? (b/e): b

Lehenengo bektorea:
...
Bigarren bektorea:
...

Ebakidura: ...

Berriro hasi nahi duzu? (b/e): e
Programa bukatu da.



2. ARIKETA: HITZ- LOTERIA(3 puntu)

Erabiltzaileak sartutako hitz zerrenda batetik, programak ausazko hitz bat aukeratu ostean, erabiltzaileak programak zein aukeratu duen igarri beharko duen jokoa da. Honako hau da jokoaren funtzionamendua:

1. Programaren hasieran, programak erabiltzaileari hitz zerrenda bat sartzeko eskatuko dio. (zenbat eta hitz gehiago sartu orduan eta zailagoa izango da jokoa).
2. Behin hitz zerrenda irakurrita, programak ausazko hitz bat hautatzen du.
3. Jarraian programak hitz bat eskatzen dio erabiltzaileari eta ausazko hitza den ala ez egiaztatzen du.
4. Programak jarraitzen du erabiltzaileak ausazko hitza zein den igarri arte edo erabiltzaileak amore eman arte.

Jokoa programatzeko *string.h* fitxategian definitutako funtzioak erabil daitezke. Gainera, ondoren aipatzen diren funtzioak definitu eta erabili behar dira:

- a) **hitzak_irakurri**: erabiltzaileari zurienez banandutako hitz zerrenda bat sartzeko eskatuko dio. **(0,25 puntu)**
- b) **hitzak_zenbatu**: karaktere-kate bat emanda, karaktere-katean dauden hitz-kopurua itzuliko du. Jakinik, hitzak zuriune batez edo gehiagoz bananduta egon daitezkeela eta karaktere-katearen hasieran (lehenengo hitza baino lehen) edo bukaeran (azkenengo hitzaren ostean) zuriuneak egon daitezkeela. **(0,5 puntu)**
- c) **igarren_hitza_lortu**: **zerrenda** izeneko karaktere-kate bat eta **i** zenbaki bat emanda, beste karaktere-kate batean gordeko du **zerrenda**-ko **i**-garren hitza (ikus beheko adibidea). Suposatzen da (ez da egiaztatu behar) **i** izango dela 1 eta hitz kopuruaren artekoa. **(puntu 1)**
- d) **berdinak_dira**: karaktere-kate bi emanda, karaktere-kate biak berdinak izanez gero 1 eta ezberdinak izanez gero 0 itzuliko du. **(0,5 puntu)**
- e) **programa_nagusia**: aurreko funtzio horiek zuzen edo zeharka erabiliko ditu, gehienez 250 karaktere dituen karaktere-kate bat eskatu, sartutako hitzetatik ausazko hitz bat aukeratu, eta erabiltzaileari igarri araziko dion programa idatzi. Jokoa amaituko da erabiltzaileak ausazko hitza zein den igartzen duenean edo sarrera huts bat sartzen duenean, amore emanez. Programak ausazko hitz bat aukeratzeko, *int ausazko_zenbaki_aukeratu(int max)* funtzioa erabiliko du. Funtzio honek 1 eta max arteko zenbaki bat itzuliko du eta *nirea.h* fitxategian inplementatuta dago, beraz EZ da programatu behar. **(0,75 puntu)**



Hurrengo adibidean, hitz-kopurua 4 da eta programak 3. hitza aukeratzen du (etxe):

' '	'e'	'a'	' '	't'	'x'	'o'	'r'	'i'	' '	'e'	't'	'x'	'e'	' '	' '	't'	'x'	'o'	' '	' '	'\0'	'?'	'...'	'?'
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	...	250

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

Hitz zerrenda bat sartu (gehienez 250 karaktere): ea txori etxe txo

Hitz bat sartu: txo

Ez duzu igarri.

Hitz bat sartu: txori

Ez duzu igarri.

Hitz bat sartu: etxe

Igarri duzu, zorionak!

Bukatzeko edozein tekla sakatu...



3. ARIKETA: ONGARRI BIOLOGIKOAK (3 puntu)

Ongarri biologikoko fabrika batean ontziratze makina batek produktuak tobera batean jasotzen ditu. Ontziratu behar den produktuaren arabera tobera zulo desberdinekin konektatzen da. Makinak ontziratzea egin ondoren zakuak zinta garraiatzaile batean uzten ditu banaketa-biltegiara joan daitezela. Aldi berean, zaku bakoitza betetakoan, bere pisua kilogramotan eta bere ontziratze-denbora erregistratzen dira fitxategi batean. Denbora horrek ez luke 9 segundo zakuko gainditu beharko.

Fitxategian erregistratutako datuak analizatu nahi dira ea anomaliak gertatu diren jakiteko, hau da, 9 segundo baino gehiagoko ontziratze-denborarik dagoen jakiteko. Anomalia bakoitzeko, zakuaren pisua libra bihurtuta eta ontziratze-denbora gorde behar dira beste fitxategi batean; fitxategi hori Ingalaterrako bulego nagusira bidaliko da.

Prozesu honen pausu guztiak burutzen dituen programa bat egin nahi da:

1. Sarrera fitxategiko datu guztiak irakurri, pisuak kilogramotan array batean eta ontziratze-denborak beste array batean gorde, soilik 9 segundoko ontziratze-denbora gainditu duten zakuentzat. Sarrera fitxategiak ez du inoiz izango 100 anomalia baino gehiago.
2. Jarraian, taulan gordetako pisuak libra bihurtzen dira eta anomalia fitxategian pisua (libra bihurtuta) eta ontziratze-denbora idazten dira.

Sarrera fitxategiaren adibidea (lerro bakoitzean zaku baten pisua eta bere ontziratze-denbora):

Pisuak	12.05	4	Denborak
	11.97	10	
	11.04	3	
	15.02	9	
	17.05	2	
	10.45	17	
	10.40	3	
	14.46	11	
	12.20	3	

Deskribatutako programa eraikitzeko, honako funtzio hauek inplementatu eta erabili behar dira:

- a) **fitxategi_izena_eskatu**: erabiltzaileari fitxategi baten izena eskatzen dio eta karaktere kate batean gordetzen du. (0.25 puntu)
- b) **datuak_kargatu**: sarrera fitxategi baten izena, zenbaki errealezko **pisuak** array-a eta osoko zenbakizko **denborak** array-a emanda, sarrera fitxategiko datuak irakurtzen ditu eta **pisuak** eta **denborak** array-etan 9 segundoko ontziratze-denbora gainditzen duten zakuen pisuak eta denborak gordetzen ditu. (puntu 1)

pisuak array-aren adibidea sarrera fitxategiko datuak irakurri ostean:

11.97	10.45	14.46	?	?
0	1	2	3	99

denborak array-aren adibidea sarrera fitxategiko datuak irakurri ostean:

10	17	11	?	?
0	1	2	3	99



- c) **libra_bihurtu**: kilogramotan adierazitako pisuak gordetzen dituen zenbaki errealezko array bat emanda, gordetako balioak libratan adierazitakoekin ordezkatzeko (kilogramotatik libratara pasatzeko: $\text{librak} = \text{kilogramoak} * 2.2046$). (0.25 puntu)

pisuak array-aren adibidea pisuak libratara pasatu ondoren:

26.39	23.04	31.88	?	?
0	1	2	3	99

- d) **datuak_idatzi**: irteera fitxategi baten izena (anomalia-fitxategia), zenbaki errealezko **pisuak** array-a eta osoko zenbakizko **denborak** array-a emanda, array-etan gordetako balioak idazten ditu irteera fitxategian. (puntu 1)

Irteera fitxategiaren adibidea:

26.39	10
23.04	17
31.88	11

- e) **programa nagusia**: aurreko funtzioak zuzenean edo zeharka erabiliz, erabiltzaileari sarrera eta irteera fitxategien izenak eskatzen dizkio eta lehen deskribatutako anomalia detektatze eta erregistratze prozesua gauzatzen du. (0,5 puntu)

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

1. Urratsa: Pisuak eta ontziratze denborak kargatzea
Fitxategiaren izena sartu:
Pisuak-eta-denborak.txt

Datuak kargatu dira

2. Urratsa: Ingalaterrako bulego nagusira bidaltzeko anomalia fitxategia sortzea
Fitxategiaren izena sartu:
Anomaliak.txt

Anomalia fitxategia sortu egin da

Bukatzeko edozein tekla sakatu...