

Oinarrizko Programazioa – Azterketa Partziala 1.

2018ko azaroaren 9

1.1 Espezifikatu eta inplementatu ezazu Python lengoaia erabiliaz zenbaki_bakoiti_mendia izeneko azpiprograma. Azpiprograma honek, sarrera bezala digitu kopuru bakoitia eta gutxienez 3 digitu dituen zenbaki bat izango du eta erdian dagoen digitua, zenbakia osatzen duten digituen artetik handiena den ala ez itzuliko du **(0.12 puntu)**.

Adibidez: 165 eta 3769572 zenbaki bakoiti mendiak dira.

Aurreko azpiprograma inplementatzeko, jada eginda dauden ondorengo azpiprogramak erabili ditzakezu (ez dira inplementatu behar):

```
def digituaI (zenb,i):  
    #aurre: zenb>100 eta 3<=digitu kopuru ditu  
    #post: digiI itzuliko du, non zenb zenbakiaren i posizioan dagoen digitua  
    #den.  
    #Adibidez: 53 zenbakiaren lehen digitua 5 da eta bigarrena 3  
  
def zenbatDigitu (zenb):  
    #post: zenb zenbakiak dituen digitu kopurua itzultzen du.
```

1.2. Espezifikatu eta inplementatu ezazu Python lengoaia erabiliaz: BaturaBakoitiMendia non, zenbaki sekuentzia bat emanda, zerrendako zenbaki bakoiti mendi guztien batura ea baita ere zenbaki bakoiti mendia den adierazten duen, hau da TRUE baiezko kasuan eta FALSE ezezko kasuan **(0.13 puntu)**.

Adibidez, sarrerako zenbaki zerrenda hau izanda <162, 1331, 4, 121, 356> egiazkoa dela esango du, zeren 162 eta 121 zenbaki bakoiti mendiak dira eta zenbaki hauen batura 283, baita ere zenbaki bakoiti mendia da, beraz TRUE itzuli beharko luke.

Ariketa honen inplementazioan, aurreko ariketan garatutako azpiprograma erabili beharko da.

Oinarrizko Programazioa – Azterketa Partziala 1.

2018ko azaroaren 9

2 Implementatu ezazu Ada lengoaia erabiliaz ondorengo azpiprograma (0.25 puntu). Ondorengo datu motak izanik:

```
bizilagun_kopurua: constant integer:= 10000;
solairu_kopurua: constant integer:= 100;
type T_Pertsona is record
    Izena: String(1..20);
    Solairua: Positive;
    Hizkia: Character;
end record;
type T_Komunitatea is array (1..bizilagun_kopurua) of T_Pertsona;
type T_EtxeOrratz is array (1..solairu_kopurua, 'A'..'J') of Integer;
```

Etxebizitza_bakoitzeko_bizilagunak_lortu azpiprograma garatu, non komunitateko informazio guztia emanik, etxe orratzeko distribuzioa lortuko duen, honako espezifikazioa izanik:

```
procedure Etxebizitza_bakoitzeko_bizilagunak_lortu (K: in T_Komunitatea;
                                                    E: out T_EtxeOrratz) is
--aurre: komunitateko bizilagunen datuak ditu K aldagaiak eta guztiz beteta
dago.
--post: K aldagaiko datuen arabera, E aldagaiak zenbat bizilagun dauden
etxebizitzako izango du.
```

Adibidez, 1^o solairuko A hizkian 5 pertsona bizi badira, komunitateko zerrendan 5 pertsonen informazioa egongo da solairu eta hizki horretan bizi direla adieraziz. Komunitateko zerrenda guztia aztertu ondoren, E (T_EtxeOrratz motako) aldagaiak, 5 balioa izango du aurreneko gelaxkan, hau da E(1, 'A')=5.

1	2	3	4	...	32	33	...	145	...	1345	...	10000
Jon	Aiora	Koldo	Miren		Mireia	Leire		Iker		Aritz		Ane
1	3	2	2		1	99		1		1		1
'A'	'B'	'A'	'A'		'A'	'A'		'A'		'A'		'A'

	'A'	'B'	...	'J'
1	5			
2	2			
3		1		
4				
...				
99	1			
100				

OHARRA: ariketa ebazteko soilik komunitateko zerrenda guztia behin zeharkatzea saia zaitetz.