

ARIKETAK

1 Algoritmo Bat Zer Den Ikusteko

- n , m , eta tmp izeneko hiru aldagai erabiliz, jarrai iezaiozu azpian datorren instrukzio sekuentziari. Hasieran n eta m aldagaiek 102 eta 30 balioak dituztela suposatu.

n	102
m	30
tmp	

■ n m -gatik zatigarri ez den bitartean

- $tmp \leftarrow (n/m)$ ren ondarra
- $n \leftarrow m$
- $m \leftarrow tmp$

- Instrukzio sekuentzia amaitu ondoren, ba al dakizu zer kalkulatu duzun m aldagaian?
- Errepika ezazu hasierako n eta m -ren balioak alderantzikoak izanik.

- n , m , $emaitz$ eta $handiena$ izeneko hiru aldagai erabiliz, jarrai iezaiozu azpian datorren instrukzio sekuentziari. Hasieran n eta m aldagaiek 12 eta 15 balioak dituztela suposatu.

n	12
m	15
$emaitz$	
tmp	

- $(n < m)$ bada,
 - $tmp \leftarrow m$
 - $m \leftarrow n$
 - $n \leftarrow tmp$
- $emaitz \leftarrow n$
- $emaitz$ m -gatik zatigarri ez den bitartean
 - $emaitz \leftarrow (emaitz + n)$

- Instrukzio sekuentzia amaitu ondoren, ba al dakizu zer kalkulatu duzun $emaitz$ aldagaian?
- Errepika ezazu hasierako n eta m -ren balioak alderantzikoak izanik.
- Eta idatzitako algoritmoa hurrengoagatik ordezkatzeko bagenu, baliagarri litzateke?

- $emaitz \leftarrow n$
 - $emaitz$ m -gatik zatigarri ez den bitartean
 - $emaitz \leftarrow (emaitz + n)$

2 Oinarrizko Sententziei Buruzkoak

1. Erabiltzaileari 3 zenbaki eskatu eta hauek txikitik handira ordenatuta pantailaratuko dizkion programa.
2. Erabiltzaileari n zenbaki arrunta eskatu eta honen arabera hurrengo motako karratua marraztuko duen programa.

n=3	n=4	n=5	n = ..
***	****	*****	
***	****	*****	
***	****	*****	
	****	*****	

3. Erabiltzaileari n zenbaki arrunta eskatu eta honen arabera hurrengo motako hirukia marraztuko duen programa.

n=3	n=4	n=5	n = ..
*	*	*	
**	**	**	
***	***	***	
	****	****	

4. Erabiltzaileari n zenbaki arrunta eskatu eta honen arabera hurrengo motako karratua marraztuko duen programa.

n=2	n=4	n=5	n = ..
*	*	*	
***	***	***	
	*****	*****	
	*****	*****	

5. Erabiltzaileari n zenbaki arrunta eskatu eta Fibonacci-ren segidako n . terminoa kalkulatu duen programa.

$$f_n = \begin{cases} 0 & \text{if } n = 0 \\ 1 & \text{if } n = 1 \\ f_{n-1} + f_{n-2} & \forall n \geq 2 \end{cases}$$

6. Erabiltzaileari n eta m bi zenbaki eskatu eta n^m kalkulatu dion programa.
7. Erabiltzaileari n zenbaki arrunta eskatu eta $n!$ kalkulatu dion programa.
8. Erabiltzaileari zenbaki arrunt bat eskatu eta hau lehena den ala ez esango dion programa.
9. Erabiltzaileari zenbaki arrunt bat eskatu eta hau perfektua den ala ez esango dion programa. Zenbaki arrunt bat perfektua dela esango dugu baldin eta soilik baldin bera baino bere zatitzaile txikiago guztien baturaren berdina bada. Adibidez: $6 = 1 + 2 + 3$.
10. Erabiltzaileari zenbaki arrunt bat eskatu eta bere karratuaren digitu guztien batuta pantailaratu dion programa.
11. Erabiltzaileari zenbaki arrunt bat eskatu eta kapikua den ala ez esango dion programa.
12. Erabiltzaileari zenbaki arrunt bat eta digitu bat eskatu eta digitu hori zenbakiko zein posiziotan dagoen esango dion programa. Posizio bat baino gehiagotan agertzen bada, lehendabizikoa erakutsiko dio soilik.

3 Funtzioei Buruzkoak

1. Errepika itzazu 2 ataleko ariketak hauetan eskatzen den eginkizun nagusirako funtzio egokia definituz. Adibidez,
 - Dimentsio emanda karratu bat marrazten duen funtzioa
 - Fibonacci-ren segidako termino baten indizea emanda termino hori itzultzen duen funtzioa
 - zenbaki arrunt bat emanda lehena den ala ez esango digun funtzioa
 - ...
2. Erabiltzaileari 2 zenbaki arrunt eskatu eta hauen *z.k.h.* eta *m.k.t.* kalkulatu eta pantailaratuko dituen programa. Eragiketa bakoitzerako funtzio bat definitu.
 - *z.k.h.* kalkulatzeko 1 ataleko 1go algoritmoari jarraitu.
 - *m.k.t.* kalkulatzeko hurrengo propietatea aplikatu.

$$mkt(n, m) = \frac{n.m}{zkh(n, m)}$$

3. Erabiltzaileari zenbaki bat eskatu eta berau baino txikiagoak diren zenbaki kapikua guztiak pantailaratuko dituen programa. Horretarako hurrengo funtzioak egin eta erabili behar dituzue:
 - **kapikuaA1Da**: zenbaki bat kapikua den ala ez itzuliko digu.
 - **itzulbiratu**: zenbaki bat emanda bere alderantzizkoa itzuliko duena. Adibidez 12340 emanda 4321 itzuliko du.

4 Bektoreei Buruzkoak

Atal honetako ariketetan, besterik ez bada esaten, erabiltzaileari zenbaki zerrenda bat eskatzerakoan zenbaki bat ilarako eskatuko diogu eta 0 zenbakiak zerrendaren amaiera markatuko du; O ez da zerrendan sartuko.

1. Erabiltzaileari zenbaki zerrenda bat eskatu eta zerrendako txikiena lehenarekin posizioz aldatu ondoren zerrenda berbera pantailaratuko duen programa.
2. Erabiltzaileari zenbaki errealen zerrenda bat eskatu eta berauen batezbestekoa, batezbestekoa baino handiagoak diren zenbaki kopurua eta batezbestekoa baino txikiagoak diren zenbaki kopurua esango dizkion programa.
3. Erabiltzaileari zenbaki zerrenda bat eskatu eta zerrenda berbera txikitik handira ordenatu ondoren berriz pantailaratuko dion programa (Honetarako erabilgarria izan daiteke 1 galderako algoritmoan oinarritu eta ariketa honetara egokitzea/zabaltzea).

4. Erabiltzaileari zenbaki zerrenda bat eskatu eta zenbaki errepikatuak kendu ondoren berriz zerrenda pantailaratuko duen programa. Amaitu aurretik, hasierako zerrenda ere pantailaratuko beharko du. Honetarako, hurrengo funtzioak egin eta erabiltzea gomendatzen zaizu:

```

/*-----
   bOrg array-ko elementuak bDest arrayra kopiatutzen dituen
   funtzioa; baina bOrg-n behin baino gehiagotan dauden
   elementuak behin bakarrik kopiatuko ditu. Adibidez:
       1 2 4 5 1 3 4 badu bOrg-g bDest 1 2 4 5 3 edukinarekin utziko du
   bDest-en gorde diren elementu kopurua itzuliko du.
   -----*/
int erreplikatuGabeKopiatu(float bOrg[], int dim, int bDest[]);

/*-----
   b array-an hasiera eta (amaiera-1) posizioen bitartean
   elementu parametroko balioa dagoen esango digun funtzioa
   -----*/
int bilatuTartean(float b[], int hasiera, int amaiera, float elementu);

```

5. Jarraian datorren kodigoa exekutatzera koan (zerrendaEskatu eta zerrendaPantailatatu funtzioak teoriako apunteetakoak dira),
- Zein da pantailaratuko duen zerrenda erabiltzaileak {12,13,23} zerrenda idatzi badu?
 - Zenbat aldagai sortu eta deuseztatu dira eta zein unetan? Ez hartu kontutan main-en argc eta argv.
 - Zein arrisku dauka exekuzio garaian?

```

int main(int argc, char* argv[])
{
    float vf1[10], vf2[10], vf3[10];
    int n,m,i;

    bektoreaHasieratu(vf1, 4);
    bektoreaHasieratu(vf3, 10);
    n=zerrendaEskatu(vf2);
    m=bektorealtsatsi(vf3, 0, vf2, n);
    m=bektorealtsatsi(vf3, n, vf1, 4);
    zerrendaPantailatu(vf3, m);
}

void bektoreaHasieratu(float v[], int dim)
{
    int i;

    for(i=0;i<dim;i++) v[i]=i;
}

int bektorealtsatsi(float v1[], int dim1, float v2[], int dim2)
{
    int i;

    for(i=0;i<dim2;i++) v1[i+dim1]=v2[i];
    return (dim1+dim2);
}

```

6. Erabiltzaileari zenbaki zerrenda bat eskatu eta zenbat elementu behin baino gehiagotan dauden esango dion programa. Adibidez, 1, 2, 8, 1, 2, 7, 7, 6, 9, 1 sekuentzia bada, 3 elementu daudela behin baino gehiagotan esan behar digu (1, 2 eta 7 hain zuzen; baina zeintzuk diren ez du esan behar).
7. Erabiltzaileari zenbaki zerrenda bat eskatu eta elementurik errepikatuena zein den esango dion programa. Elementu ugari baldin badaude kopuru berean, beraietako bat esatea aski da. Adibidez, zerrenda 1 2 3 1 2 3 1 2 3 4 5 1 2 bada, 1 eta 2 zenbakiak dira gehien errepikatuta dudenak; baina kopuru berean. Kasu honetan 1 zenbakia dela esatea aski da.

5 Karaktere Kateei Buruzkoak

1. Erabiltzaileari esaldi bat eskatu eta esaldi berbera letra larriez soilik pantailaratuko duen programa (letrak ez diren karaktereak berdin utzi behar ditu)
2. Erabiltzaileari esaldi bat eskatu eta zenbat hitzez dagoen osatuta esango dion programa. Hitza, hutsuneak ez diren edozein karaktere sekuentzia dela suposatuko dugu
3. Erabiltzaileari esaldi bat eskatu eta esaldi berbera hitz artean hutsune bakarra dagoelarik pantailaratuko duen programa. Hasierako eta amaierako hutsuneak ere kendu egin beharko ditu
4. Idatzi zerorrek `strlen` funtzioaren baliokide bat eta berau probatzeko programa “*simple*” bat. Emaiozu `gureStrlen` izena.
5. Idatzi zerorrek `strcpy` funtzioaren baliokide bat eta berau probatzeko programa “*simple*” bat. Emaiozu `gureStrcpy` izena.
6. Idatzi zerorrek `strcat` funtzioaren baliokide bat eta berau probatzeko programa “*simple*” bat. Emaiozu `gureStrCpy` izena.
7. Idatzi zerorrek `strcmp` funtzioaren baliokide bat eta berau probatzeko programa “*simple*” bat. Emaiozu `gureStrcmp` izena.
8. Erabiltzaileari esaldi bat eta beste karaktere sekuentzi bat eskatu (bigarren hau hitz bat izan daiteke) eta sekuentzi hau hasierako esaldiaren barne badagoen ala ez esango digun programa.
9. Erabiltzaileari esaldi bat eta beste karaktere sekuentzi bat eskatu (bigarren hau hitz bat izan daiteke) eta sekuentzi hau hasierako esaldian zenbat aldiz dagoen esango dion programa.