3. GAIA. INFORMAZIOA DATU EGITURETAN GORDE

0.- AURKIBIDEA

0.- AURKIBIDEA 1

01.- ARRAYAK. DEFINIZIOA 2

02.- ARRAY DEKLARAZIOA. SASIKODEA. 3

03.- ARRAY DEKLARAZIOA. JAVA. 4

04.- ARRAY BAT ARAKATZEA. 5

05.- ARRAY BATEN ELEMENTU BERRIA SARTZEA. 6

06.- ARRAY BATEKO ELEMENTU BAT EZABATU 7

07.- ZENBAKIZKO ARRAYAK. ARIKETAK. SASIKODEA. JAVA 8

08.- KARAKTEREZ EDO STRINGEZ BETETAKO ARRAYAK 10

09.- STRING-EKIN LANEAN 11

10.-STRING BERDINKETA KONPARATZEN 12

11.- STRING MOTAKO DATUAK KONPARATZEN 13

12.- ZENBAKIZKO BALIOA STRING BIHURTU 14

13.- STRING BALIOA ZENBAKIZKOA BIHURTU 15

14.- KARAKTERE ETA STRINGEZKO ARRAYEN ARIKETAK 16

15.- ZENBAKIZKO MATRIZEAK 17

16.- MATRIZEEN HASIERATZEA 18

17.- MATRIZEAK ARAKATU 19

18.- ZENBAKIZKO MATRIZEEN ARIKETAK 20

19.- STRINGEZKO ARRAYA 21

20.- ARRAYAK ORDENATZEKO ARIKETAK 22

21.- ARRAYETAKO ARIKETA OSOAK 23

01.- ARRAYAK. DEFINIZIOA

Array bat mota bereko aldagaien bilduma da. Aldagai horiek bata bestearen ondoan daude, eta, erakusle motako aldagai baten bidez erabili ditzakegu. Erakusle horrek arrayaren lehenengo posizioa seinalatzen du. Array bat gordetzeko behar den memoria kantitatea honako hauen araberakoa da: elementuen mota, elementuen tamaina eta arrayaren tamaina.

Indize baten bidez helduko gara arrayaren edozein elementutara; indize hori 0tik *N*-1era doa, non *N* elementuen kopurua den. Array bateko elementuak atzitzean, tentuz egin beharko dugu indizeak arrayaren azken posizioa gaindi ez dezan handia; izan ere, posizio hori arraytik kanpokoa izango litzateke, eta ez genuke izango bertara joateko baimenik.

02.- NOLA DEKLARATU ARRAYAK. SASIKODEA.

Array bat deklaratzeko, arrayaren datuen mota, izena eta tamaina adierazi behar ditugu.

Eman dezagun bost ikasleren notak jasotzeko array bat deklaratu nahi dugula. Honelakoa izango litzateke deklarazioa sasikodean:

OSOA ARRAYNOTAK[5];

03.- NOLA DEKLARATU ARRAYAK. JAVA.

Javan, lehenengo eta behin, arraya deklaratu behar dugu. Gero, sortu egingo dugu *new* funtzioa erabiliz.

Honela deklaratuko genuke aurreko ataleko arraya Javan:

int arraynotak[ ];

arraynotak= new int[5];

04.- ARRAY BAT ARAKATU

Array bat arakatzea bere elementuak banan-banan atzitzea da. Arraya hasieratzeko, balio jakin bat bilatzeko… izan daiteke baliagarria.

05.- ARRAY BATEAN ELEMENTU BERRI BAT SARTU.

Array batean elementu berri bat sartu baino lehen, beterik ez dagoela egiaztatu behar dugu. Beterik balego, tamaina handitu beharko genioke.

Elementu berri bat sartzeko, aukera asko ditugu:

**Sartu amaieran :** erabilitako azken posizioaren hurrengoan jarri beharko dugu elementu berria.

Eman dezagun array hau daukagula:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 |
| 2 | 4 |  |  |  |

*Urdinez, indizeak.*

Orain 5 balioa gehitu nahi badiogu gehitu nahi dugu arrayaren bukaeran. Horretarako, informazio hau beharko dugu: arrayaren tamaina eta erabilitako azken posizioa. Elementu berria sartutakoan, erabilitako posizioak 3 izango dira; hortaz, komeni zaigu posizio horiek jakiteko aldagai bat ere izatea (*erabiliak* aldagaia, adibidez).

*indizea=5*

*erabiliak=3*

**Sartu posizio jakin batean :** elementu berria sartu baino lehen, beteko duen posizioaren atzeko elementu guztiak eskuinera mugitu beharko ditugu.

Eman dezagun 1 balioa sartu nahi dugula lehenengo posizioan (0). Aldez aurretik, eskuinera mugitu beharko ditugu beste balio guztiak.

*indizea=5*

*erab=4*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 |
| 1 | 2 | 4 | 5 |  |

**Sartu ordena errespetatuz:** Txertatze metodoa erabiliko dugu. Lehenengo, elementuari dagokion posizioa bilatu; gero, posizio horren atzeko elementu guztiak birkolokatu; eta, azkenik, elementu berria txertatu beharko dugu.

Eman dezagun 3 elementua sartu nahi dugula arrayan:

*indizea=5*

*erab=5*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 |
| 1 | 2 | 3 | 4 | 5 |

Lehenengo, array guztia arakatu beharko dugu, dagokion posizioa bilatzeko (2). Gero, atzeko elementuak mugitu (4 eta 5), eta, azkenik, eguneratu egingo ditugu indizeak *(erab=5)*.

06.- ARRAY BATEN ELEMENTU BAT EZABATU

Hona array baten elementu bat ezabatzeko bi teknika:

**Ezabatu elementua, eta markatu**: balio bat pentsatu behar dugu ezabatu nahi dugun elementuei esleitzeko —elementuek sekula izango ez duten balio bat—. . Gero, balio hori daukan elementua ezabatu, eta antolatu egingo dugu arraya.

Eman dezagun 4 balioa ezabatu nahi dugula. Zenbaki osoen arraya denez, -1 balioa emango diogu dagokion elementuari. Gero, 5 balioa ezkerrerantz mugituko dugu.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 |
| 1 | 2 | 3 | -1 | 5 |

Dena dela, metodo hau ez da batere egokia, eta ez dugu erabiliko.

**Ezabatu arraya trinkotuz**: metodo hau posizioa dakigunean zein ezezaguna dugunean erabili daiteke; azken kasu horretan, lehenengo pausoa elementuaren posizioa jakitea izango da—. Edonola ere, posizio horren osteko elementu guztiak posizio bat ezkerrera mugitu beharko ditugu.

Eman dezagun 4 elementua ezabatu nahi dugula. Lehenengo, posizioa *(3)* lortu, eta, gero, 3 posiziora eramango dugu 5 balioa.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 |
| 1 | 2 | 3 | 5 |  |

Ondoren, *erabiliak* aldagaiaren balioa eguneratuko dugu.

*indizea=5*

*erabiliak=5*

07.- ZENBAKIZKO ARRAYAK. ARIKETAK. SASIKODEA. JAVA

1.- Egin *ZENBATZNBK* programaren sasikodea. Programa horrek, 1etik 5erako zenbakiak eskatu, eta zera esango du: zenbat aldiz sartu den zenbaki bakoitza; zein den gehien sartutakoa; zein den gutxien sartutakoa; eta zer zenbaki ez diren behin ere sartu. Jasotako zenbakia 0 denean, amaitu egingo da programa.

2.- Egin *BATEZBESTE* programaren sasikodea. Programa horrek, teklatutik zenbakiak eskatu, eta array batean sartuko ditu erabiltzaileak erabaki arte (zenbaki negatiboa sartuz), edo arraya bete arte. Gero, batezbestekoa kalkulatuko du programak. Ondoren, hauek erakutsiko ditu:

* Zenbat zenbaki dauden batezbestekotik gorago, eta zenbat beherago.
* Lagina.

3.- Egin *BATAZBESTETENP* programaren sasikodea. Programa horrek, teklatutik tenperaturak eskatu, eta array batean sartuko ditu erabiltzaileak erabaki arte (zenbaki negatiboa sartuz), edo arraya bete arte. Gero, batezbesteko tenperatura kalkulatuko du, eta esan egingo du tenperatura bakoitza batezbestekoa baino handiagoa edo txikiagoa bataz bestekoa baino handiagoa edo txikiagoa —edo berbera—den.

4.- Egin *ZENBATZNBK* programa. Programa horrek, 1etik 5erako zenbakiak eskatu, eta zera esango du: zenbat aldiz sartu den zenbaki bakoitza; zein den gehien sartutakoa; zein den gutxien sartutakoa; eta zer zenbaki ez diren behin ere sartu. Jasotako zenbakia 0 denean, amaitu egingo da programa.

5.- Egin BATAZBESTE programa. Programa horrek, teklatutik zenbakiak eskatu, eta array batean sartuko ditu erabiltzaileak erabaki arte (zenbaki negatiboa sartuz), edo arraya bete arte. Gero, batezbestekoa kalkulatuko du programak. Ondoren, hauek erakutsiko ditu:

* Zenbat zenbaki dauden batezbestekotik gorago, eta zenbat beherago.
* Lagina.

6.- Egin BATAZBESTETENP programa. Programa horrek, teklatutik tenperaturak eskatu, eta array batean sartuko ditu erabiltzaileak erabaki arte (zenbaki negatiboa sartuz), edo arraya bete arte. Gero, batezbesteko tenperatura kalkulatuko du, eta esan egingo du tenperatura bakoitza batezbestekoa baino handiagoa edo txikiagoa bataz bestekoa baino handiagoa edo txikiagoa —edo berbera—den.

7.- Egin *IZENABANAN* programa. Programa horrek, teklatutik izena eskatu, eta banaka idatziko ditu letrak.

Sartu zure izena: *KEPA*

K

E

P

A

8.- Egin *IZENABANANALDE* programa. Programa horrek, teklatutik izena eskatu, eta banaka idatziko ditu letrak, baina alderantziz.

Sartu zure izena: *KEPA*

A

P

E

K

9.- Egin *LETRAKALDATU* programa. Programa horrek, izena, posizioa eta letra bat eskatu, eta jasotako letra berria jarriko du posizio horretako letraren ordez.

Sartu izena: kepa

Posizioa: 2

Letra berria: a

Emaitza: kapa

10.- Egin *IZENAALDATU* programa.. Programa horrek, izena eskatu, eta alderantziz idatziko du.

Sartu zure izena: kepa

Alderantziz: apek

11.- Egin *IZENATXIKIHANDI* programa. Programa horrek, izena eskatu, eta, erabiltzaileak aukeratuta, letra larriz edo letra xehez jarriko du.

Sartu izena: *Kepa*

Handiz edo txikiz(h/x): *h*

*KEPA*

12.- Egin *IZENAORDENATU* programa. Programa horrek, 3 izen eskatu, eta ordenatu egingo ditu alfabetikoki.

13.- Egin *ALDAGAIMOTALDATU* programa. Programa horrek, zenbaki bat eskatu, eta string bihurtuko du.

14.- Egin *ALDAGAIMOTALDATU2* programa. Programa horrek, string bat eskatu, eta integer *int* bihurtuko du.

08.- KARAKTEREZKO ARRAYAK ETA STRINGAK

Javan, karakterezko arrayak eta stringak ez dira gauza bera. Honela deklaratzen dira string-ak eta karakterezko arrayak, hurrenez hurren:

*String katea=””;*

*String katea= new String();*

Horrela, string nulua izango genuke. Hasierako balioa eman nahi badiogu, honela egin behar dugu:

*String katea=”Kaixo mundua”;*

*String katea= new String(“Kaixo mundua”);*

09.- STRING-EKIN LANEAN

Ezin dira eragiketa berberak egin stringekin eta karakterezko arrayekin. Karakterezko arrayekin eta zenbakizko arrayekin eragiketa berberak egin daitezke.

String batek karakterezko array gisa jokatzea nahi baldin badugu, hau egin beharko dugu: stringaren edukia sartu karakterezko array batean.

Hona hemen stringekin lan egiteko hainbat eragiketa:

**Lortu stringaren posizio jakin bateko karakterea:** *charAt()* metodoa erabiliko dugu.

Eman dezagun stringaren 2 karakterea nahi dugula:

*karakterea= charAt(2);*

**Jarri karaktere bat stringaren posizio jakin batean**. Lehenengo stringa array bihurtu, gero, balioa aldatu, eta, azkenik, arraya string bihurtuko dugu berriro ere.

Eman dezagun ‘A’ karakterea jarri nahi dugula stringaren lehen posizioan:

*string katea=”kaixo mundua”;*

*char[] kateatenp=katea.toCharArray();*

*kateatemp[0]=’A’;*

*katea=new String (kateatemp);*

**Jarri letra larriz String baten karaktere guztiak:** *toUpperCase()* metodoa erabili beharko dugu.

Eman dezagun *katea* stringeko karaktere guztiak letra larriz jarri nahi ditugula:

*katea=katea.toUpperCase();*

**Jarri letra xehez String baten karaktere guztiak:** *toLowerCase()* metodoa erabili beharko dugu

Eman dezagun *katea* stringeko karaktere guztiak letra xehez jarri nahi ditugula:

*katea=katea.toLowerCase();*

**Lortu string baten luzera:** *Length()* metodoa erabiliko dugu.

Eman dezagun *katea* stringaren luzera lotu nahi dugula:

*katea=katea.Length ();*

10.- NOLA KONPARATU STRINGAK

Ezin da berdintasun operadorea (==) erabili. Equals metodoa erabili behar da.

Berdintasun-eragilea (==)

Berdinketa operadorea (==) honetarako erabiltzen da: string motako bi aldagaik objektu berberari erreferentzia egiten dioten jakiteko. Kasu honetan, bi aldagaiek objektu berari egingo diote erreferentzia:

*String s1=”kaixo mundua”;*

*String s2=s1;*

Beraz, (s1==s2) konparazioak egia (True) itzuliko du.

String motako bi aldagaik balio berdina duten jakiteko, aldiz, *equals* erabiltzen da.

*String s1=”Kaixo mundua”;*

*String s2=new String(“Kaixo mundua”);*

*If (s1.equals(s2));* Egia

*If (s1==s2);* Gezurra

String motako datuak alfabetikoko konparatzeko, *compareTo(string)* metodoa erabiltzen da.

*s1.compareTo(s2)*

Itzulitako zenbakia 0 baino txikiagoa baldin bada, *s1<a2* izango da.

Itzulitako zenbakia 0 baino txikiagoa baldin bada, *s1>a2* izango da.

Itzulitako zenbakia 0 baldin bada, *s1=a2* izango da.

*String s1=”Kaixo mundua”; String s2=”Agur mundua”*

*A=s1.compareTo(s2); A>0*

*String s1=”Kaixo mundua”; String s2=”Txintxo ibili”*

*A=s1.compareTo(s2); A<0*

*String s1=”Kaixo mundua”; String s2=”Kaixo mundua”*

*A=s1.compareTo(s2); A=0*

12.- ZENBAKIZKO BALIOA STRING BIHURTU

Batzuetan, zenbakizko balio bat string bihurtu nahi dugu. Horretarako, metodo hau erabiliko dugu: *String.valueOf(balioa)*. Funtzio horrek zenbakizko balio osoa(int, long…) edo erreala( double, float..) hartu, eta kate bihurtzen du.

*String s=String.valueOf(5); s=”5”*

*String pi=String.valueOf(3.14) pi=”3.14”*

13.- STRING BALIOA ZENBAKIZKO BIHURTU

Beste batzuetan, string balio bat zenbakizko bihurtu nahi dugu. Lehenengo eta behin, egokia da hasierako eta bukaerako zuriuneak kentzea, *trim* metodoa erabiliz.

Gero, stringa zenbaki oso bihurtzeko, *Integer.parseInt(string)* metodoa erabiliko dugu.

*String s = “5”; int n=Integer.parseInt(s.trim());* n=5

Stringa zenbaki erreal bihurtzeko, aldiz, bi pauso egin behar ditugu: hasteko, string-a double motako objektu bihurtuko dugu *Double.valueOf(string)* metodoarekin; ondoren, objektua *double* bihurtuko dugu *doubleValue()* erabiliz**.**

*String s=”3.14”*

*Double D=Double.valueOf(s.trim());*

*Double d=D.douleValue();*

14.- KARAKTERE ETA STRINGEZKO ARRAYEN ARIKETAK

15.- Egin *PASAHITZA* klasea. Klase horrek izena eta pasahitza eskatuko ditu teklatutik. 10 erabiltzaile-pasahitza izango ditu gordeta array batean, eta erabiltzaileak sartutakoak bertan bilatuko ditu. Baldin badaude, sartzen utziko dio; bestela, ez, eta amaitu egingo da programa . Hiru saiakera izango ditu gehienez erabiltzaileak.

16.- Egin *KARAKNAGUSIA* klasea. Klase horrek karaktere katea eskatu, eta zera esango digu: zein karaktere den handiena eta zein txikiena, bietan ere posizioa aipatuz.

17.- Egin *KONTABOK* klasea. Klase horrek, karaktere kate bat eskatu, eta bokal bakoitza zenbat aldiz agertzen den esango du.

18.- Egin *BETEMATRIZEA01* klasea. Klase horrek, erabiltzaileak sartutako 12 zenbakirekin 4x3ko zenbakizko matrizea bete, eta gero antolaturik aurkeztuko ditu.

*Sartu zenbaki bat: 1*

*….*

*Hau da matrizea*

*1 4 7*

*3 3 8*

*7 9 9*

*5 3 2*

19.- Egin *BILATUZNBK* klasea. Klase horrek, erabiltzaileari zenbaki bat eskatu, eta aurreko ariketako matrizean bilatuko du, posizioa ere aipatuz.

*Sartu zenbakia: 9*

*Zenbakia 3,3 posizioan dago.*

15.- ZENBAKIZKO MATRIZEAK

Matrize bat bi dimentsiotako arraya da. Errenkadak izango ditu, eta errenkadetan zutabeak. Izan, beraz, array bat da, edo, hobe esanda, arrayez osaturik dago: errenkada bakoitzean array bat sortzen da, eta array horretan hurrengo arrayaren helbidea gordetzen da.

Hau izango litzateke 3 errenkada eta 2 zutabeko matrize baten itxura :

*Matrizea zutabea 0 zutabea 1*

*Errenkada 0 1 2*

*Errenkada 1 5 7*

*Errenkada 2 8 2*

Memorian, honela egongo litzateke kokatua:

|  |  |  |
| --- | --- | --- |
| *E0* | *E1* | *E2* |

*Matrizea*

|  |  |  |
| --- | --- | --- |
| *E0* | *1* | *2* |
| *E1* | *5* | *7* |
| *E2* | *8* | *2* |

Javan matrize bat definitzeko, bere datu mota adierazi behar dugu lehenengo; gero, dimentsioak (*[][]*); eta, amaitzeko, izena.

Eman dezagun matrize bat definitu nahi dugula *Int* motako datuez osatua. *matrizea* izena izango du, eta 3 errenkada eta 2 zutabe izango ditu.

*int [][] matrizea= new int[3][2];*

Errenkaden zutabe kopuruak ez dira berdinak izan behar.

*Int [][] matrizea=new int[3][];*

*matrizea [0] =new int[2]; /\*Errenkada honek 2 zutabe ditu\*/*

*matrizea [1] =new int[3]; /\*Errenkada honek 3 zutabe ditu\*/*

*matrizea [2] =new int[4]; /\*Errenkada honek 4 zutabe ditu\*/*

Horrelako matrizeak, hala ere, ez dira oso ohikoak.

16.- MATRIZEAK HASIERATU

Matrize bat egitean, konpiladoreak balio bat ematen dio posizio bakoitzari. Balio hori motaren araberakoa da.

Adb:

Datua zenbakizkoa baldin bada, 0 balioa.

Datua karaktere bat baldin bada, **‘/u0000’** balioa: karaktere nulua.

Datua boolearra baldin bada, **False** balioa

Datua string bat baldin bada,**null** balioa

Datua beste edozein motatakoa baldin bada, **null** balioa

Matrize bat hasieratzeko, bere errenkada eta zutabeen kopurua jakin behar ditugu. Errenkada guztiek zutabe kopuru berbera baldin badute, begizta baten bidez hasieratu dezakegu.

Adb:

Honako hau egingo dugu 3 errenkada eta 2 zutabeko matrize bat hasieratzeko:

*for (int erren=0,erren<3; erren++)*

*{*

*for (int zut=0;zut<2; zut++)*

*{*

*matrizea[erren][zut]=-1; //-1 ekin betetzen dut matrizea*

*}*

*}*

Sartu beharreko datuen motak ezberdinak baldin badira edo zutabe kopurua aldakorra baldin bada, beste metodo bat ere erabili dezakegu:

Honako hau egingo dugu 3 errenkada eta 2 zutabeko matrize bat sortu, eta hasierako balio hauek esleitzeko: 1,2,3,4,5,6

*Int [][] matriz={{1,2},{3,4},{5,6}};*

Eman dezagun 3 errenkadako matrize bat sortu nahi dugula. Bi zutabe ditu 0 errenkadan; hiru 1 errenkadan; eta lau 2 errenkadan. Hauek dira balioak: {1,2},{3,4,5},{6,7,8,9}

*Int [][] matriz={{1,2},{3,4,5},{6,7,8,9}};*

17.- MATRIZEAK ARAKATU

Matrize bat arakatzeko, bere errenkada eta zutabe kopurua jakin behar ditugu. Errenkada guztietan zutabe kopuru bera baldin badauka, begizta bat erabili ahal izango dugu.

Eman dezagun 3 errenkada eta 2 zutabeko matrize bat arakatu, eta eduki guztia pantailatik atera nahi dugula. Honela egin genezake:

*for (int erren=0,erren<3; erren++)*

*{*

*for (int zut=0;zut<2; zut++)*

*{*

*system.out.println(matrizea[erren][zut]);*

*}*

*}*

Matrizeak ez badauka errenkada guztietan zutabe kopuru bera, berriz, length metodoa erabili behar dugu.

Eman dezagun 3 errenkadako matrize bat sortu nahi dugula. Bi zutabe ditu 0 errenkadan; hiru 1 errenkadan; eta lau 2 errenkadan. Honela egin genezake:

*for (int erren=0,erren<3; erren++)*

*{*

*for (int zut=0;zut<****matrizea.[erren].length****; zut++)*

*{*

*system.out.println(matrizea[erren][zut]);*

*}*

*}*

Posible da *length* metodoa errenkadetan zein zutabeetan erabiltzea:

*for (int erren=0,erren<****matrizea.length****; erren++)*

*{*

*for (int zut=0;zut<****matrizea.[erren].length****; zut++)*

*{*

*system.out.println(matrizea[erren][zut]);*

*}*

*}*

18.- ZENBAKIZKO MATRIZEEN ARIKETAK

20.- Egin *ARRAYBATU* izeneko programa bat. Programa horrek, teklatutik 20 zenbaki eskatu, eta, errenkada bakoitzeko, elementu guztien batura gordeko ditu beste array batean. Bukatzeko, zenbakiak eta batura agertaraziko ditu pantailan.

21.- Egin *HASI2D* izeneko programa bat. Programa horrek, bi dimentsioko matrize bat beteko du *#* karakterearekin. Matrizea 7X4koa izango da.

22.- Egin *IRAULI* izeneko programa bat. Programa horrek, 3\*4ko matrize baten elementu guztiak eskatu, eta beste matrize bat sortuko du —bestearen iraulia—. Gero, matrize biak agertuko dira pantailan.

23.- Egin *MATBATU* izeneko programa bat. Programa horrek, 3\*2ko matrize bi eskatu, eta bien batuketa egingo du. Ondoren, pantailan idatziko du emaitza.

24.- Egin *MATKEN* izeneko programa bat. Programa horrek, 3\*2ko matrize bi eskatu, eta bien kenketa egingo du. Ondoren, pantailan idatziko du emaitza.

25.- Egin *ARRAYERAGIKETA* izeneko programa bat. Honako menu hau aurkeztuko du:

ARRAYEKIN ERAGIKETAK:

1.- Arraya hasieratu.

2.- Arraya ikusi.

3.- Txertatu elementu bat.

4.- Elementu bat ezabatu.

5.- Irten.

26.- Egin *BIDERMAT* izeneko programa bat. Programa horrek, bi matrize eskatu, eta biderkatu egingo ditu. Horretarako, bien zutabe kopurua berbera izan behar da. Metodo hau erabili beharko dugu biderketa egiteko.

A Matrizea B matrizea

A11 A12 A13 B11 B12 B13

A21 A22 A23 B12 B22 B32

A31 A32 A33 B31 B32 B33

A41 A42 A43

Biderketa

A11\*B11+A12\*B21+A13\*B31 A11\*B12+A12\*B22+A13\*B32 A11\*B13+A12\*B23+A13\*B33

A21\*B11+A22\*B21+A23\*B31 A21\*B12+A22\*B22+A23\*B32 A21\*B13+A22\*B23+A23\*B33

A31\*B11+A32\*B21+A33\*B31 A31\*B12+A32\*B22+A33\*B32 A31\*B13+A32\*B23+A33\*B33

A41\*B11+A42\*B21+A43\*B31 A41\*B12+A42\*B22+A43\*B32 A41\*B13+A42\*B23+A43\*B33

19.- STRINGEZKO ARRAYA

Stringezko arrayek zenbakizkoek bezala funtzionatzen dute.

Eman dezagun 5 karaktere-kate gorde nahi ditugula array batean. Honela egin genezake:

*String[]=new String[5];*

*String katea;*

20.- ARRAYAK ORDENATZEKO ARIKETAK

27.- Egin *ARRAYORD* programa. Programa horrek, teklatutik zenbaki positiboak eskatu, eta ordenatuta sartuko ditu array batean. Programa zenbaki negatibo bat jasotzean bukatuko da.

28.- Egin *HIZTEGI* programa. Programa horrek, teklatutik 20 hitz eskatu, eta alfabetikoki ordenatuko ditu.

29.- Egin *HIZTEGIALD* programa. Programa horrek, teklatutik 20 hitz eskatu, eta alfabetikoki ordenatuko ditu, baina beherantz.

30.- Egin *BATERATU* program. Programa horrek, 10 hitz eskatu, eta alfabetikoki ordenatuko ditu. Gero, gauza bera egingo du beste array batekin. Azkenik, beste array batean bateratuko ditu biak .

21.- ARRAYEN DENETARIKO ARIKETAK

31.- Egin *MULTALEA* izeneko programa bat. Programa horrek, 10 posizioko bi array egin, eta ausazko zenbakiz beteko ditu. Ondoren, posizio bereko elementuak biderkatu, eta emaitza beste array batean utziko du.

33.- Egin *NAN* izeneko programa bat. Programa horrek, NAN bat eskatu, eta beraren hizkia ondo dagoela egiaztatuko du.

34.- Egin *KAPIKUA* izeneko programa bat. Programa horrek, 5 zifrako zenbaki bat eskatu, eta kapikua denetz esango du.

35.- Egin *ZERRENDA* izeneko programa bat. Programa horrek 5 ikasleren abizena, izena eta 3 nota —Sareak, Databaseak eta Programazioa— eskatuko ditu. Ondoren, datu hauek aterako ditu pantailan:

Ikasleko:

Batez bestekoa.

Zenbat *gutxi* eta zein ikasgaitan.

Nota altuena eta baxuena —eta zein ikasgaitan—.

Ikasgaiko:

Batez bestekoa.

Nota altuena eta nork atera duen.

Nota baxuena eta nork atera duen.

*Gutxi*en kopurua.

Guztira:

Batez bestekoa.

Nota altuena, nork atera duen, eta zein ikasgaitan.

Nota baxuena, nork atera duen, eta zein ikasgaitan.

Zein izan den *gutxi* gehien atera duen ikaslea.

Zein izan den *gutxi* gehien izan duen ikasgaia.