**Tablouri unidimensionale. Probleme de înlocuire, formare, interschimbări, eliminări, inserări și sortări ale elementelor.**

**Modificarea sau înlocuirea elementelor unui tablou unidimensional**

Algoritmul ce stă la baza tuturor acestor probleme constă din trei etape:

1. Completarea sau formarea tabloului;
2. Verificarea elementelor tabloului și modificarea lor;
3. Afișarea tabloul modificat pe ecran.

Fiecare dintre aceste etape este realizată într-un ciclu separat, dar a doua etapă (adică verificarea și modificarea tabloului) poate fi combinată fie cu prima (intrare), fie cu a treia (ieșire).

***Exemplul 1***

Fie dat un tablou unidimensional de numere întregi. Înlocuiți toate elementele negative care au indice impar cu valoarea lor absolută.

**Intrare:**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| -4 | -3 | -10 | 1 | -5 | 0 | -1 | -1 | -6 | 8 |

**Ieșire:**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| -4 | 3 | -10 | 1 | -5 | 0 | -1 | 1 | -6 | 8 |

**Program**

**#include** <iostream>

**#include** <cmath>

**#include** <ctime>

**#include** <cstdlib>

**using namespace** std;

**int** main()

{

srand(time(0));

**int** i,n=10,a[n];

cout<<endl;

**for** (i=0; i<n; i++)

{

a[i]=-10+rand()%20;

cout<<a[i]<<" ";

}

cout<<endl;

**for** (i=0; i<n; i++)

**if** (a[i]<0 && i%2!=0)

a[i]=abs(a[i]);

**for** (i=0; i<n; i++)

cout<<a[i]<<' ';

**return** 0;

}

***Exemplul 2***

Fie dat un tablou unidimensional de numere întregi. Adăugați un element cu indicele **k** la toate elementele tabloului care sunt mai mici de **10**.

**Rezultatul programului**

|  |  |
| --- | --- |
| **Intrare** | **Ieșire** |
| k=5  12 3 4 44 7 6 22 16 7 10 | 12 10 11 44 14 13 22 16 14 10 |

**Problema se complică** prin aceea că elementul **a[k]** poate fi și el mai mic ca **10**, deci poate fi modificat și apoi acest element modificat va fi adăugat elementelor de după elementul din poziția **k**. Pentru a preveni acest lucru, elementul din poziția **k** este memorizat într-o variabilă temporară.

**Program**

**#include** <iostream>

**#include** <cmath>

**#include** <ctime>

**#include** <cstdlib>

**using namespace** std;

**int** main()

{

srand(time(0));

**int** i,n=10,a[n],k,aux;

**for** (i=0; i<n; i++)

{

a[i]=rand()%30;

cout<<a[i]<<" ";

}

cout<<"dati k="; cin>>k;

aux=a[k];

**for** (i=0; i<n; i++)

**if** (a[i]<10)

a[i]=a[i]+aux;

cout<<endl;

**for** (i=0; i<n; i++)

cout<<a[i]<<' ';

**return** 0;

}

***Exemplul 3***

Fie dat un tablou unidimensional cu elemente de tip caracter. Înlocuiți fiecare caracter **'a'** cu caracterul **'!'**.

**Intrare:**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| f | h | a | a | y | i | e | a | y | a |

**Ieșire:**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| f | h | ! | ! | y | i | e | ! | y | ! |

**#include** <iostream>

**using namespace** std;

**int** main()

{

**const int** n=10;

**char** a[n];

**for** (**int** i=0; i<n; i++)

cin>>a[i];

**for** (**int** i=0; i<n; i++)

{

**if** (a[i]='a')

a[i]='!';

cout<<a[i]<<" ";

}

**return** 0;

}

***Sarcini de lucru***

1. Fie dat un tablou unidimensional de numere întregi. Dublați toate elementele sale.
2. Fie dat un tablou unidimensional de numere întregi. Înlocuiți toate elementele cu indice impar cu **0**.
3. Fie dat un tablou unidimensional de caractere. Înlocuiți toate elementele cu indice par cu caracterul **'\*'**.
4. Fie dat un tablou unidimensional de numere reale. Pentru fiecare element cu valoare mai mare decât **10**, obțineți rădăcina pătrată.
5. Fie dat un tablou unidimensional de numere reale. Incrementați toate elementele cu indice par cu **1**, iar cei cu indice impar – decrementați cu **1**.
6. Fie dat un tablou unidimensional de numere întregi. Micșorați în jumătate toate elementele care se termină cu cifra **4**.
7. Fie dat un tablou unidimensional de numere reale. Măriți toate elementele cu indice impari cu numărul **k**.
8. Fie dat un tablou unidimensional de numere reale. Din toate elementele pozitive scădeți valoarea elementului cu indicele **k1** și din celelalte – valoarea elementului cu indicele **k2**.
9. Fie dat un tablou unidimensional de numere întregi. Descreșteți toate elementele impare cu **m** și creșteți elementele cu indice impar cu **k**.
10. Fie dat un tablou unidimensional de caractere. Înlocuiți toate literele **'q'** cu literele **'w'**.
11. Fie dat un tablou unidimensional de numere întregi. Înlocuiți toate numerele mai mari de **10** cu numărul **10**.
12. Fie dat un tablou unidimensional de numere reale. Măriți toate elementele cu indice impar cu valoarea numărului **k**.
13. Fie dat un tablou unidimensional de numere reale. Înlocuiți primele **k** elemente cu cele opuse lor.
14. Fie dat un tablou unidimensional de numere întregi. Înlocuiți fiecare element aflat în poziție impară cu pătratul indicilor lor.
15. Fie dat un tablou unidimensional de numere reale. Împărțiți toate elementele tabloului cu indice par la primul element (dacă nu este **0**).
16. Fie dat un tablou unidimensional de numere întregi. Înmulțiți toate elementele tabloului care sunt multipli de **3** cu al treilea element.
17. Este dat un tablou unidimensional de numere întregi. Scădeți elementul cu indicele **k1** din toate elementele pozitive și adăugați elementul cu indicele **k2** la toate elementele negative.
18. Este dat un tablou unidimensional de numere întregi. Scădeți elementul cu indicele **k** din toate elementele pozitive și adăugați numărul **N** la toate elementele negative.
19. Fie dat un tablou unidimensional de numere întregi. Înlocuiți toate elementele până la primul număr par cu **0**.
20. Este dat un tablou unidimensional de numere reale distincte. Înlocuiți cu **0** toate elementele tabloului situate între elementul maxim și minim.

**Formarea unui tablou unidimensional**

Această fișă de laborator conține următoarele tipuri de sarcini:

1. Din elementele tabloului **A** obțineți elementele tabloului **B** cu același index.
2. Din elementele tabloului **A** obțineți elementele tabloului **B** cu un index diferit.
3. Divizarea unui tablou unidimensional în mai multe tablouri;
4. Combinarea mai multor tablouri într-un singur tablou.

***Exemplul 1***

Fie dat un tablou unidimensional **A**, care conține **n** numere întregi. Din elementele tabloului **A** formați un alt tablou unidimensional **B** de aceeași dimensiune, după următoarea regulă: dacă indicele **i** al elementului tabloului **A** este par, atunci , în caz contrar .

**Intrare:**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
| **7** | **8** | **4** | **1** | **3** | **7** | **5** | **9** | **2** | **5** |

**Ieșire:**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
| **49** | **16** | **16** | **2** | **9** | **14** | **25** | **18** | **4** | **10** |

Acesta este primul tip de problemă, care poate fi rezolvată după algoritmul următor:

1. Introduceți date în tabloul **A**;
2. Parcurgeți elementele tabloului **A**, formați în același timp tabloului **B** din elementele lui **A** și afișați elementele tabloului **B** pe ecran.

**#include** <iostream>

**using namespace** std;

**int** main()

{

**int** n=10;

**int** a[n], b[n];

**for** (**int** i=0; i<n; i++)

cin>>a[i];

**for** (**int** i=0; i<n; i++)

{

**if** (i%2==0)

b[i]=a[i]\*a[i];

**else**

b[i]=2\*a[i];

cout<<b[i]<<' ';

}

**return** 0;

}

***Exemplul 2***

Fie dat un tablou unidimensional **A**, care conține **n** numere întregi. Din elementele acestui tablou se formează un tablou B de aceeași dimensiune în care sunt situate mai întâi elementele pare ale tabloului A și apoi cele impare.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Intrare:** | **5** | **3** | **7** | **8** | **6** | **3** | **1** | **2** | **0** | **5** |
| **Ieșire:** | **8** | **6** | **2** | **0** | **5** | **3** | **7** | **3** | **1** | **5** |

Aceasta este o sarcină de al doilea tip, cu două parcurgeri ale tabloului **A**, conform algoritmului:

1. Introduceți date în tabloul **A**;
2. Parcurgeți elementele tabloului **A**, selectați elementele pare și scrieți-le în tabloul **B**;
3. Parcurgeți din nou tabloul **A**, selectați elementele impare din **A** și adăugați-le la tabloul **B**;
4. Afișați tabloul **B** pe ecran.

**#include** <iostream>

**using namespace** std;

**int** main()

{

**int** n=10, a[n], b[n];

**for** (**int** i=0; i<n;i++)

cin>>a[i];

**int** j=0;

**for** (**int** i=0;i<n;i++)

**if** (a[i]%2==0)

{

b[j]=a[i];

j++;

}

**for** (**int** i=0;i<n;i++)

**if** (a[i]%2==1)

{

b[j]=a[i];

j++;

}

**for** (**int** i=0;i<n;i++)

cout<<b[i]<<" ";

**return** 0;

}

***Exemplul 3***

Fie dat un tablou unidimensional de numere întregi **A** de dimensiunea **2\*n**. Din elementele acestui tablou se formează două tablouri **B** și **C** de dimensiunea n. În tabloul **B** rescrie elementele din tabloul **A** cu indici de la **0** la **n-1** (prima jumătate), iar în tabloul **C** rescrie elementele cu indici de la **n** la **2n-1** (cea de-a doua jumătate a tabloului **A)**.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Tabloul A** | **5** | **3** | **7** | **8** | **6** | **3** | **1** | **2** | **0** | **5** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tabloul B** | **5** | **3** | **7** | **8** | **6** |
| **Tabloul C** | **3** | **1** | **2** | **0** | **5** |

**Algoritm**

Această problemă este de tip mixt, cu următorul algoritm:

1. Introduceți datele în tabloul **A**;
2. Parcurgeți elementele tabloului **A** de la **0** la **(n-1)** și scrieți-le în tabloul **B**. Indicii elementelor tablourilor **A** și **B** coincid;
3. Parcurgeți elementelor tabloului **A** de la **n** la **(2\*n-1)** și scrieți-le în tabloul **C**. Indicii elementelor tablourilor **A** și **C** nu se potrivesc ;
4. Afișați tablourile **B** și **C** pe ecran.

**Program**

**#include** <iostream>

**using namespace** std;

**int** main()

{

**int** n=5;

**int** a[2\*n];

**int** b[n], c[n];

**for** (**int** i=0; i<2\*n; i++)

cin>>a[i];

**for** (**int** i=0; i<n; i++)

b[i]=a[i];

**int** j=0;

**for** (**int** i=n; i<2\*n; i++)

{

c[j]=a[i];

j++;

}

**for** (**int** i=0;i<n;i++)

cout<<b[i]<<" ";

cout<<endl;

**for** (**int** i=0;i<n;i++)

cout<<c[i]<<" ";

**return** 0;

}

***Exemplul 4***

Fie dat un tablou unidimensional **A** de dimensiunea **n** și un tablou unidimensional **B** de dimensiunea **m**. Din elementele acestor tablouri se formează un tablou unidimensional **C** de dimensiunea **n+m**, în care să rescrie mai întâi elementele tabloului **A** și apoi elementele tabloului **B**.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tabloul A** | **1** | **2** | **-2** | **-5** | **-3** |
| **Tabloul B** | **4** | **-1** | **-1** | **1** | **-3** |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Tabloul C** | **1** | **2** | **-2** | **-5** | **-3** | **4** | **-1** | **-1** | **1** | **-3** |

**Program**

**#include** <iostream>

**#include** <ctime>

**#include** <cstdlib>

**using namespace** std;

**int** main()

{

srand(time(0));

**int const** n=5, m=5;

**int** i,j;

**int** a[n],b[m],c[n+m];

**for** (i=0;i<n;i++)

{

a[i]=-5+rand()%11;

cout<<a[i]<<" ";

}

cout<<"\n";

**for** (i=0;i<m;i++)

{

b[i]=-5+rand()%11;

cout<<b[i]<<" ";

}

**for** (i=0;i<n;i++)

c[i]=a[i];

j=n;

**for** (i=0;i<m;i++)

{

c[j]=b[i];

j++;

}

cout<<"\n";

**for** (i=0;i<n+m;i++)

cout<<c[i]<<" ";

**return** 0;

}

***Sarcini de realizat***

1. Fie date două tablouri unidimensionale **A** și **B** de numere reale cu același număr de elemente **n**. Formați un tablou unidimensional **C**, fiecare element al căruia fiind egal cu maximumul elementelor din aceeași poziție din tablourile **A** și **B**.
2. Fie dat un tablou unidimensional **P** de numere reale. Din elementele acestui tablou formați un alt tablou unidimensional **Q** de aceeași dimensiune după regula: elementele cu indicele **i** de la **3** la **8** se obțin prin formula **Qi=-Pi**, celelalte – prin formula **Qi=Pi\*i**.
3. Din elementele tabloului **M** formați un tablou unidimensional **N** de aceeași dimensiune după următoarea regulă: dacă numărul de ordine **i** al elementului tabloului **M** este par, atunci **Ni=i\*Mi**, în caz contrar **Ni=Mi/i**.
4. Din elementele tabloului **P** formați un tablou unidimensional **Q** de aceeași dimensiune după regula: elementele cu numerele **i** de la **3** la **10** se obțin după formula **Qi=Pi+i**, restul – după formula **Qi=Pi-i**.
5. Din elementele tabloului **A**, formați tabloul **B** de aceeași dimensiune conform regulii: dacă elementele tabloului **A** sunt pare, atunci le dublați, cele impare se copiază fără schimbări.
6. Din elementele tabloului **M**, formați un tablou unidimensional **N** de aceeași dimensiune după următoarea regulă: elementele nenegative ale tabloului **M** se reduc de trei ori, cele negative – se ridică la pătrat.
7. Sunt date două tablouri de numere întregi **A** și **B** de aceeași dimensiune. Formați un al treilea tablou **C** de aceeași dimensiune, în care fiecare element se obține după următoarea regulă: **Ci=Ai\*Bi**.
8. Fie date două tablouri de numere întregi **A** și **B** de aceeași dimensiune. Formați un al treilea tablou **C** de aceeași dimensiune, fiecare element al căruia fiind media aritmetică a elementelor corespunzătoare din tablourile **A** și **B**.
9. Fie date două tablouri de aceeași dimensiune **T1** și **T2**, în care nu există elemente nule. Formați tabloul **T3**, fiecare element al căruia fiind egal cu **1**, dacă elementele tablourilor date cu același indice au aceleași semne și este egal cu **0** în caz contrar.
10. Fie date două tablouri de numere întregi **P** și **Q**, de dimensiuni diferite. Formați un al treilea tablou **R**, în care elementele celui de-al doilea tablou sunt situate la început, iar apoi elementele primului tablou.
11. Fie dat un tablou unidimensional de numere întregi. Rescrieți elementele sale într-un alt tablou de aceeași dimensiune, după cum urmează: mai întâi toate elementele negative din tabloul dat, apoi celelalte elemente.
12. Din tabloul **M** în tabloul **N** rescrieți mai întâi elementele pozitive, apoi cele nule și apoi elementele negative.
13. Din tabloul **M** în tabloul **N** rescrieți mai întâi elementele multiplu de 2, apoi multipli de 3, dar nu multipli de 2 și în final restul elementelor.
14. Este dat un tablou unidimensional de numere reale alcătuit din zece elemente. Formați două tablouri de dimensiunea **5**,incluzând în primul tablou elementele cu indice impar din tabloul dat, iar în al doilea tablou – cele cu indice par.
15. Este dat un tablou unidimensional de numere reale. Rescrieți elementele sale pozitive în a doua tablou, iar celelalte elemente în al treilea tablou.
16. Fie date două tablouri de numere reale **A** și **B** de dimensiunea **n**. Formați tabloul **C** de aceeași dimensiune, fiecare element fiind format conform următoarelor legități: **C0=A0+Bn-1**, **C1=A1+Bn-2**, **C2=A2+Bn-3,...**., **Cn-1=An-1+B0**.
17. Fie date două tablouri de numere reale **A** și **B** de dimensiunea **2\*n**. Formați tabloul **C** de dimensiunea **n**, fiecare element al cărui fiind format prin formulele: **C0=A0\*B1**, **C1=A1\*B3**, **C2=A2\*B5**, ..., **Cn-1=An-1\*B2\*n-1**.