masivebi

- I. erTganzomilebiani masivi: A={a_0 , a_1 , . . ., a_n }

A

a_0	a_1	. . .	. . .	a_n

II. organzomilebiani masivi: A=

$$\begin{matrix} a_{00} & \dots & a_{0n} \\ \dots & \dots & \dots \\ a_{m0} & \dots & a_{mn} \end{matrix}$$

		. . .	. . .	
		. . .	. . .	
		. . .	. . .	

erTganzomilebiani masivis gamocxa-deba da elementebiT Sevseba

***masivis tipi da saxeli* [*elem. raodenoba*] ;**

mag. int b[15], i;

```
for(i=0;i<15;i++)
cin>>b[i];
```

analogiurad xdeba gamotana:

```
for(i=0;i<15;i++)
cout<<b[i]<<“   “;
```

SemTxveviTi ricxvebis generireba

funqcia rand ( ) realizebulia cstdlib biblioTekaSi. miiReba ricxvi [0; Rand_Max-1] diapazonidan. Rand_Max =32767.

rand()%n-s SemTxvevaSi diapazoni iqneba [0;n-1];

rand()%n-k SemTxvevaSi diapazoni iqneba [-k;n-k-1];

rand()%n+k SemTxvevaSi diapazoni iqneba [k;n+k -1];

mag. d=rand()%17-5; ricxvi iqneba [-5; 11] diapazonidan.

funqcia srand.

zogadi saxiT Caiwereba srand (argumenti); misi daniSnulebaa rand() funqciisaTvis sawyisi monacemis gamomuSaveba.

srand(time(0)); - argumentad aqvs mimdinare dro kompiuterSi. uzrunvelyofs programis yoveli gaSvebisas gansxvavebuli SemTxveviTi ricxvis miRebas.

time(0) realizebulia ctime biblioTekaSi.

```
# include<iostream>
# include<cstdlib>
# Include<ctime>
. . . . . .
srand(time(0));
b=rand();
```

ერთგანზომილებიანი მასივის (ვექტორის) ელემენტების შეტანა/გამოტანა

```
#include <iostream>
using namespace std;
int main()
{
double a[15], s=0,sa;
int i,k=0;
  for (i=0;i<15;i++)
  cin>>a[i];
cout<<endl<<endl;
  for (i=0; i<15; i++)
  cout<<a[i]<<"   ";
  cout<<endl;
for(i=0;i<15;i++)
```

mag.1. mocemuli $C=\{c_0,c_1, \ldots ,c_n\}$ $n\leq 25$ veqtoris udidesi mniSvn-bis mqone elementis povna.

```
# include <iostream>
using namespace std;
int main ()
{
double c[26], max; int n, i;
cin>>n;
for (i=0 ; i<n ; i++)
cin>>c[i] ;
max=c[0];
for (i=1 ; i<n ; i++)
```

mag.2. B={b_0,b_1,. . . ,b_{14}} veqtoris dadebiTi elementebis jamis gamoTvla.

```
# include <iostream>
using namespace std;
main ()
{
double b[15], sum; int i;
for (i=0 ; i<15 ; i++)
cin>>b[i] ;
sum=0;
for (i=0 ; i<15 ; i++)
```

mag.3. $C=\{c_0,c_1, \ldots ,c_k\}_{k\le 22}$ veqtoris aranulovani elementebis saSualo ariTmetikulis gamoTvla.

```
# include <iostream>
using namespace std;
main ()
{
double c[23], sum=0,sa;
int i,n=0,k;
cin>>k;
for (i=0 ; i<k ; i++)
cin>>c[i] ;
```

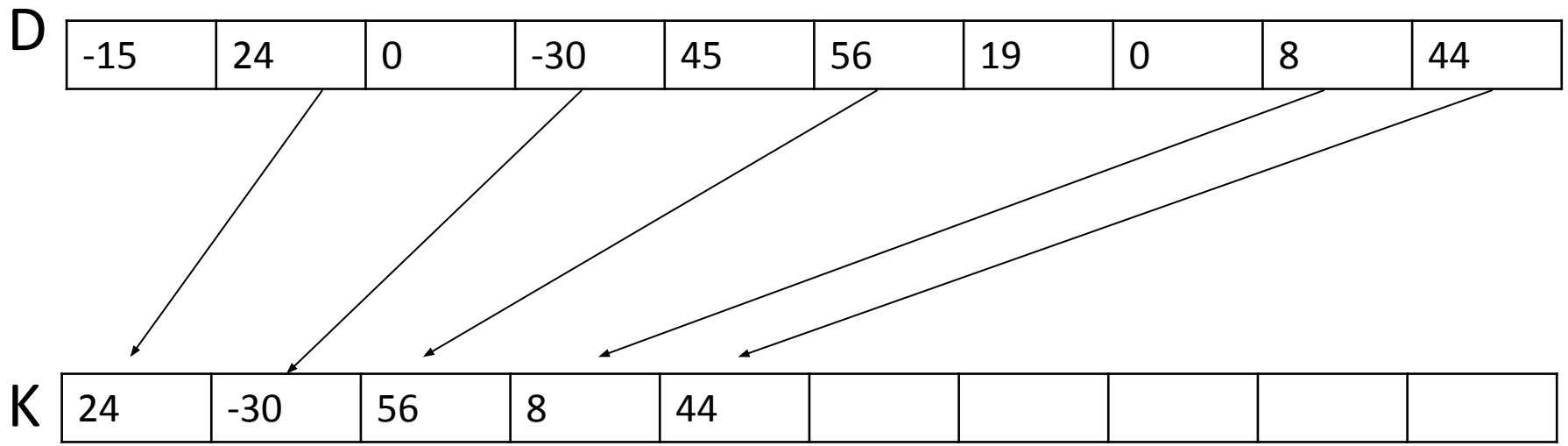
mag.4. $C=\{c_0,c_1,\ldots,c_n\}_{n\leq 24}$ veqtoris dadebiT elementebs Soris umciresi mniSvnelobis mqone elementis rigiTi nomris gamoTvla.

```
# include <iostream>
using namespace std;
main ()
{
double c[25], min=1e+06;
int i,k,n;
cin>>n;
for (i=0 ; i<n ; i++)
cin>>c[i] ;
```

mag.5. $D=\{d_0,d_1, \ldots ,d_n\}_{n\le 24}$ veqtoris luwindeq-siani, nulisagan gansxvavebuli elementebis namravlis gamoTvla.

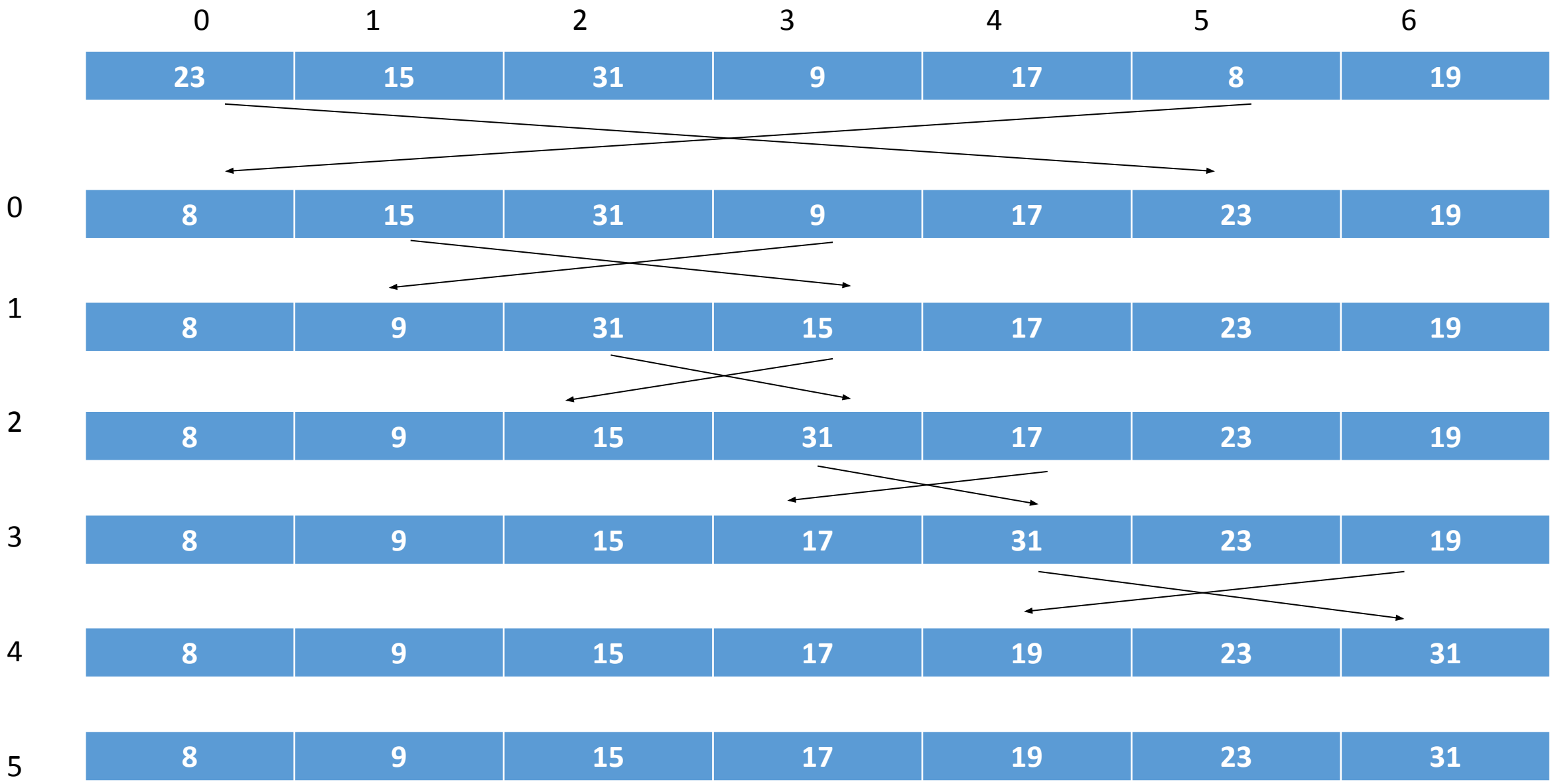
```
# include <iostream>
using namespace std;
int main ()
{
double d[25], p=1; int i,n;
cin>>n;
for (i=0 ; i<25 ; i++)
cin>>d[i] ;
for (i=2 ; i<25 ; i+=2)
if (d[i]>0) p*=d[i];
```

mag.6. mocemuli $D=\{d_0,d_1, \ldots ,d_n\}_{n\le 15}$ veqtoris luwi mniSvnelobis mqone elementebisagan axali $K=\{k_0,k_1, \ldots ,k_n\}_{n\le 15}$ veqtoris miReba da gamotana ekranze.



```
# include <iostream>
using namespace std;
int main ()
{
int d[15],k[15], i,j=0,n;
cin>>n;
for (i=0 ; i<n ; i++)
cin>>d[i] ;
for (i=0 ; i<n ; i++)
if (d[i]!=0 && d[i]%2==0) {k[j]=d[i];j++;}
cout<<"\n\n satskisi D masivi:  "<<endl;
for (i=0 ; i<n ; i++)
cout<<d[i]<<" ";
```

მასივის დალაგება „ამორჩევის“ მეთოდით



```
# include <iostream>
using namespace std;
int main ()
{int a[15], i,j,n,min,k,r;
cin>>n;
for (i=0 ; i<n ; i++)
cin>>a[i] ;
   for (i=0;i<n-1;i++)
     {  min=a[i];k=i;
      for (j=i+1; j<n;j++)
       if(a[j]<min){min=a[j];k=j;}
       if (k!=i){r=a[k]; a[k]=a[i]; a[i]=r;}
     }
cout<<"\n\n dalagahuli A masivi:\n";
```

მასივის დალაგება “ჩაძირვის” მეთოდით

```
# include <iostream>
using namespace std;
int main ()
{ int a[15], i,j,n,i1,r;
cin>>n;
for (i=0 ; i<n ; i++)
cin>>a[i] ;
 cout<<"\n\n satskisi A masivi: "<<endl;
for (i=0 ; i<n ; i++)
cout<<a[i]<<" ";
cout<<'\n';
```