

3.5.18 Пішін минус 50 -ден 50- диапазонында 20 кездейсоқ бүтін сандардың массиві. Оны басып шығару. Массивтің 1 элементін 20 элементпен, 2-ні 19-бен және 10-мен орынын ауыстыру. Жаңа массивті басып шығарыңыз.

3.5.19 Кездейсоқ  $A ( X, Y )$  үйлестіреді және  $D ( X, Y )$  биіктерге қарама-қарсы жүз тіктөртбұрыш көрсетілген жинақталатын . 0-ден минус 150 үйлестіру мәндер массиві . ( - Квадраттар бөлек екі жақ жұп 4 тараптар және барлық тараптар қанағаттандырады онда тіктөртбұрыш санын басып шығару ) анықтау және сол тараптардың осы тіктөртбұрыш кез келген жағдайда басып шығарыңыз.

3.5. 100 тізе радиусы - кездейсоқ орталығы және  $R X$  және  $Y$  координаты жинақталатын . 150 минус 150 мәндерді үйлестіру ауқымы , 5-тен 25 радиусы мәндер диапазоны. Максимум анықтау және « салынған » топтарының ( шеңбер кейбір басқа бөлігінде , т.б. толық шеңбері ) ретін басып шығарыңыз.

### **3.6 СРСП-да есеп-хатты қорғауға арналған бақылау сұрақтары**

3.6.1 Сілтеме типі түсінігі. Мысал.

3.6.2 Массив түсінігі. Деректерді массив түрінде ұйымдастыру мысалдары.

3.6. Қандай массивтер бағдарламада жарияланады? Мысал.

3.6.4 Массивті баптандыру қалай орындалады? Мысал.

3.6.5 Кездейсоқ сандардың генераторын пайдаланып массивтерді қалай құруға болады? Мысал.

3.6.6 Массив элементтеріне ассоциативті алгоритм бойынша қатынау. Мысал.

3.6.7 Массив элементтерін әдістік таңдау бойынша сұрыптау алгоритмі. Алгоритм және код фрагментін сұрыптаудың ауызша сипаттамасы .

3.6.8 «Көпіршік» алгоритмі бойынша массив элементтерін сұрыптау. Алгоритм және код фрагментін сұрыптаудың ауызша сипаттамасы .

3.6.9 Орын ауыстыру алгоритмі (оңға немесе солға ауыстыру) деректер массивтерінде. Алгоритм және код фрагментін сұрыптаудың ауызша сипаттамасы .

3.6.10 Динамикалық массив түсінігі. Мысал.

3.6.11 Іздеу массивінің түсінігі және іздеу кілт. Мысал .

3.6.12 Массив элементтерін бірізділік іздеу алгоритмі. Бірізділік іздеу алгоритмінің артықшылығы және кемшілігі.

3.6.13 Массив элементінің блоктық іздеу алгоритмі. Блоктық іздеу алгоритмінің артықшылығы және кемшілігі.

3.6.14 Массив элементтерінің екілік іздеу алгоритмі. Екілік іздеу алгоритмінің артықшылығы және кемшілігі.

3.6.15 Адрес-хештеудің түрлену кілтінің іздеу алгоритмі.

## **ТАҚЫРЫП 4 С# ТІЛІНІҢ ӘДІС-ФУНКЦИЯСЫН ҚОЛДАНУ**

### **4.1 Төртінші тақырып мақсаты**

Класс әдістерін және мәліметтерін оқу, әдістерді пайдаланып консоль қосымшасын құруға практикалық дағдылану.

## 4.2 Теориялық мәліметтер

### 4.2.1 Әдіс түсінігі

C# тілі объектілі-бағдарланған бағдарламалау тілі болып табылады және класс оның негізі болып табылады. Класстар мәліметтер типі ретінде қарастырылады, оның компоненттері ретінде (басқалармен бірге) класс өрістерін қамтиды (оның деректері) және класс әдістері (оның функциясы). Класс әдістері "қызмет етеді" мәліметтерге, оларды өзгертумен айналысады.

C# тілінде функциялар кейбір класстардың әдістер ретінде қызмет етеді, олар класстан тыс жүреді.

C# тілінде арнайы кілттік сөздер жоқ – method, procedure, function, бірақ олар жайлы түсінік бар. Әдісті жариялау синтаксисі әдісті – процедура немес функция екенін анықтауға мүмкіндік береді.

### 4.2.2 Класс әдісінің жазу форматы

Класс әдісінің жазу форматы келесідей:

```
void немес тип_метода имя_метода(список_формальных_параметров)
{ тело метода }
```

Егер әдіс типінің орнына void мәні берілсе, онда әдіс процедура сияқты жұмыс істейді.

Егер void сөзі болмаса, орына әдіс типі көрсетілсе, онда әдіс функция ретінде жұмыс істейді.

Әдіс атауын жазу міндетті, ол формальды параметрлер тізімі емес жағдайда қажет болуы оның әдісінің түрін, әдісі аты мен жақшаларды, көрсету болып табылады. Әдіс атауы және формальді параметрлер тізімі әдістің қолын көрсетеді (бағдарламаны пайдаланған кезде қажет элементтер). Назар аударыңыз, қол қоюға формальді параметрлердің аты кірмейді, бұл жерде аргументтер аты маңызды. Қол қоюға әдіс типі де кірмейді.

Мысалы, әдіс процедурасын келесідей көрсетуге болады:

```
void poisk() {...};
```

оны бағдарламада қолдану келесідей көрсеткішпен шектеледі:

```
poisk();
```

Әдіс функциясының типін жариялағанда, мысалы:

```
int kol() {...};
```

бағдарламаны пайдалану түрі бүтін бір айнымалы берілуге тиіс (әдіс типіне сәйкес), мысалы:

```
d = kol();
```

### 4.2.3 Класс әдісінің формальді параметрлері

Әдіс параметрінің формальді тізімі бос болуы мүмкін және класс әдістері үшін қалыпты жағдай. Тізім параметрлері бекітілген сан болуы мүмкін, үтірмен бөлінген.

Бір формальді параметрдің жазу форматы:

[ref или out или params]тип\_параметраимя\_параметра

Жақшада берілген мәндер міндетті емес.

Тип пен параметр атауын көрсету міндетті болып табылады.

Ресми параметрлерге тіркелген санға қарамастан нақты параметрлерінің кез келген санына өтуге мүмкінді бар. Ресми параметрлер тізімінде осы мүмкіндікті іске асыру үшін params кілт сөзі болуы тиіс. Бұл өткен параметр тізімі түрі кез келген массив ретінде жарияланды ғана хабарландыруда пайда болуы мүмкін. Бұл ресми параметрді шақыру кезде нақты параметрлердің еркін санына сәйкес келеді.

Барлық әдіс параметрлері үш топқа бөлінеді: кіріс , шығыс және жаңартылған .

Бірінші топ параметрі әдіс ақпаратын береді, олардың мәні әдіс денесінде оқылады.

Екінші топ параметрі әдістің нәтижесін көрсетеді, әдіс жұмыс істеп тұрған кезде мән қабылдайды.

Үшінші топ параметрі екі функция атқарады. Бұл құндылықтар есептеу пайдаланылады және әдіс нәтижесінде жаңартылады. Шығыс параметрі out кілттік сөзімен бірге жүруі тиіс, жаңартылады – ref. Енгізу параметрлері не болсақ, олар кілт сөзсіз көрсетілген. Егер параметр out кілттік сөзімен шығыс ретінде жарияланса, онда әдіс денесінде міндетті түрде меншіктеу операторы болуы мүмкін, ол параметр мәнін көрсетеді. Әйтпесе, қате компиляция кезінде жүреді. Әдіс денесі операторлар тізбегін көрсетеді және айнымалыларды сипаттайды, фигуралық жақша ішінде көрсетілген. Егер сөз функция денесі жайлы болса, онда блокта кем дегенде бір оператор болуы керек, функция мәнін return<выражение> түрінде қайтарады (мән типі функция типімен бірдей болуы керек).

Әдіс денесінде көрсетілген айнымалылар сол әдіс ішінде локальді болып есептеледі.

Әдіс денесінің операторларын жазуда әдіс айнымалылардың локальді аты қолданылады, класстың өріс аттары (әдістер жаһандық айнымалылар класы ретінде қарастырылады) және әдіс параметрлері аттары.

### 4.3 Зертханалық жұмысты орындауға арналған мысал

Әдісті пайдаланып жұмыс істеу үшін таза білім беру бағдарламасын қарастырайық.

Есеп 4.1 минус 50-ден 50-ге дейінгі диапазонда кездейсоқ 20 саннан тұратын массив тұрғызу керек. Оны басып шығару. Массив мәнін 1 разрядқа солға жылжытуды орында. Тапсырманың әрбір тапсырмасын көрсететіндей бағдарламаны әдіс түрінде ұсыну. Әдіс ішінде кіріс, шығыс және жаңартылған параметрлерді қолдану. Мәзірді қарастыру.

Есептің шартына сәйкес 3 әдісті орындау қажет:

– массив құру (массив үшін шығыс параметрін қолданамыз);

– массивті басу (массив үшін кіріс параметрін қолданамыз);  
 – массивті жылжыту (массив үшін жаңартылған параметрді қолданамыз).

Бағдарламаның мәзірін ұйымдастыру үшін цикл while циклін және switch () операторын қолданамыз;

Бағдарлама коды:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace ConsoleApplication1
{
class Program
{
public static void sozd(out int[] ma)
{
    ma = new int[20];
    Random rnd = new Random();
    for (int i = 0; i < 20; i++)
        ma[i] = rnd.Next() % 101 - 50;
    Console.WriteLine("Массив создан !!");
}

public static void zadvig(ref int[] ma)
{
    int k;
    for (int i = 0; i < 19; i++)
    {
        k = ma[i]; ma[i] = ma[i + 1]; ma[i + 1] = k;
    }
    Console.WriteLine("Сдвиг массива на 1 разряд выполнен !");
}

public static void prinmas(int[] ma)
{
    for (int i = 0; i < 20; i++)
        Console.Write(" {0}", ma[i]);
    Console.WriteLine();
}

static void Main()
{
    int[] a = new int[20];
    int k = 0;
    string buf;
    while (k < 4)
    {
        Console.WriteLine("1 - Создать массив 20 чисел");
        Console.WriteLine("2 - Переместить массив на 1 разряд влево");
        Console.WriteLine("3 - Печать массива");
        Console.WriteLine("4 - Выход из программы");
    }
}
```

```

Console.WriteLine("Введите пункт меню программы");
buf = Console.ReadLine();
    k = Convert.ToInt32(buf);
switch (k)
    {
case 1: sozd(out a); break;
case 2: zadvig(ref a); break;
case 3: prinmas(a); break;
default: break;
    }
}
}
}
}

```

Бағдарлама жұмысы:

- 1 - 20 саннан тұратын массив құру
- 2 – массивті 1 рарядқа солға жылжыту
- 3 – массивті басып шығару
- 4 – бағдарламадан шығу

Бағдарламаның мәзірін енгізіңіз

1

Массив құрылды !!

- 1 - 20 саннан тұратын массив құру
- 2 - массивті 1 рарядқа солға жылжыту
- 3 - массивті басып шығару
- 4 - бағдарламадан шығу

Бағдарламаның мәзірін енгізіңіз

3

-24 -20 40 46 -26 -16 -45 -46 -39 32 38 -38 -18 -2 3 -26 -40 -17 -34 -39

- 1 - 20 саннан тұратын массив құру
- 2 - массивті 1 рарядқа солға жылжыту
- 3 - массивті басып шығару
- 4 - бағдарламадан шығу

Бағдарламаның мәзірін енгізіңіз

2

Массивті 1 разрядқа солға жылдыту орындалды!

- 1 - 20 саннан тұратын массив құру
- 2 - массивті 1 рарядқа солға жылжыту
- 3 - массивті басып шығару
- 4 - бағдарламадан шығу

Бағдарламаның мәзірін енгізіңіз

3

-20 40 46 -26 -16 -45 -46 -39 32 38 -38 -18 -2 3 -26 -40 -17 -34 -39 -24

- 1 - 20 саннан тұратын массив құру
- 2 - массивті 1 рарядқа солға жылжыту
- 3 - массивті басып шығару

4 - бағдарламадан шығу  
Бағдарламаның мәзірін енгізіңіз

#### 4.4 Зертханалық жұмысқа үй тапсырмасы

Диалог режимінде берілген  $x$  ( $x > 0$  және  $x < 1$ ) үшін әдіс-функцияны қолданып, қатардың соммасын табыңыз. Исползуя метод-функцию вычислить сумму ряда для заданного в режиме диалога  $x$  ( $x > 0$  и  $x < 1$ ). Қатардың кезекті мүшесі  $\varepsilon = 0.0001$  дәлдіктен кіші болғанша есептеуді жалғастыру керек.:

$$\sqrt[3]{1+x} = 1 + \frac{1}{3}x - \frac{2}{2! \cdot 3^2}x^2 + \frac{2 \cdot 5}{3! \cdot 3^3}x^3 - \frac{2 \cdot 5 \cdot 8}{4! \cdot 3^4}x^4 + \dots$$

#### 4.5 СРС-да орындалатын жеке тапсырма

**Назар аударыңыз! Орындалатын есептің әрбір әдіс-функциясында немесе әдіс-процедурасында кіріс және жаңартылатын формальді параметрлер болу керек. Егер осы параметрлер қолданбаса, жеке тапсырма қорғауға қабылданбайды.**

4.5.1 20 кесіндінің ( $Y_1, Y_2$ ) және ( $X_1, X_2$ ) координаттары кездесек түрде құрылады. Координаттардың мәндері – 0-ден 100 дейінгі бүтін сандар. Максимал ұзындықты кесіндінің нөмерін тап. Әдіс-процедураны қолдан.

4.5.2 Қатарды есептейтін бағдарламаны жаз:

$$Y = \sum_{N=0}^{100} (x^{2 \cdot N + 1} / (2 \cdot N + 1) \cdot \sin(2 \cdot N + 1) \cdot x / 10)$$

мұнда  $x$  – диалог режимінде енгізіледі интервалы 0 ден 1 дейін;  $N - 1$  қадаммен 0-ден 100 дейін өзгеріледі. Әдіс-функцияны қолдану керек.

4.5.3  $ax^2 + bx + c = 0$  өрнектің түбірлерін табатын бағдарламаны жаз.  $a$ ,  $b$  және  $c$  мәндері диалог режимінде енгізіледі. Өрнектің түблері бар болу шартын тексеріңіз және тиісті хабарламаны шығарыңыз. Бұл есепті шешу үшін әдіс-процедураны қолданыңыз.

4.5.4 Әдіс-функцияны қолданып  $C$ -нің мәнін есепте:

$$C_N^M = \frac{N!}{M!(N-M)!}$$

Егер  $N$  саны 0-ден үлкен болса,  $N!$  есепте, әйтпесе – экран мониторуна тиісті хабарламаны шығар.  $N$  және  $M$  мәндері кездейсоқ түрде құрылады немесе экран мониторынан енгізіледі.

4.5.5 Дөңгелек берілген, центрдің координаттары (50,50) және радиус = 30. 100 нүктенің ( $x, y$ ) координаттары кездейсоқ түрде құрылады. Координаттар мәндерінің диапазоны 0-ден 100-ге дейін. Қанша нүкте дөңгелектің ішіне кіргенін анықтаңыз. Әдіс-процедураны қолданыңыз

4.5.6  $Y = 1*2*3+2*3*4+3*4*5+\dots+(n-1)*n*(n+1)$  соммасын тап, n диалог режимінде беріледі. Әдіс-функцияны қолданыңыз.

4.5.7 Циклда диапазоны 0-ден 100-ге дейін болатын, кездейсоқ 20 бүтін сандар қалыптастырылады. Сандардың мәндерін басып шығарыңыз. max және min сандарды анықтау үшін әдіс-функцияны пайдаланыңыз. Оларды басып шығарыңыз.

4.5.8 Диалог режимінде енгізілген және  $|x| > 1$  үшін қатардың соммасын табыңыз.

$$y = \frac{2!}{x^2 * 3!} + \frac{3!}{x^4 * 4!} + \frac{4!}{x^6 * 5!} + \dots$$

Қатардың кезекті мүшесі 0.0001 мәнінен кіші болғанға дейін есептеуді жалғастыру керек. Әдіс-функцияны қолданыңыз.

4.5.9 100 нүктенің X және Y координаттары кездейсоқ түрде құрылады. Координаттар мәндерінің диапазоны -150-ден +150-ге дейін. Әрбір ширекте орналасқан нүктелердің санын экранға шығарыңыз. Ось бойында орналасқан нүктелерді жеке қарастырыңыз. Әдіс-процедураны қолданыңыз.

4.5.10  $|x| > 0$  үшін әдіс-функцияны қолдана отырып, қатардың соммасын табыңыз.

$$\ln \frac{x+1}{x-1} = 2 \sum_{n=0}^{\infty} \frac{1}{(2n+1)x^{2n+1}} = 2 \left( \frac{1}{x} + \frac{1}{3x^3} + \frac{1}{5x^5} + \dots \right) \quad |x| > 1$$

Қатардың кезекті мүшесі 0.0001 мәнінен кіші болғанға дейін есептеуді жалғастыру керек.

4.5.11 Жүз тіктөртбұрыштардың қарама-қарсы төбелері A(X,Y) және B(X,Y) кездесок түрде құрылады. Координаттар мәндерінің диапазоны -150-ден +150-ге дейін. Координаттар жүйесінің жоғарғы және төменгі бөліктерінде орналасқан тіктөртбұрыштардың санын тауып, экран мониторуна шығару керек (егер төбелері әртүрлі ширекте жатса, онда ол тіктөртбұрыш қарастырылмайды). Әдіс-процедураны қолдану керек.

4.5.12  $x = 5$  үшін әдіс-функцияны қолдана отырып, қатардың соммасын табыңыз.

$$\arctg x = \frac{\pi}{2} + \sum_{n=0}^{\infty} \frac{(-1)^{n+1}}{(2n+1)x^{2n+1}} = \frac{\pi}{2} - \frac{1}{x} + \frac{1}{3x^3} - \frac{1}{5x^5} \dots \quad x > 1$$

Қатардың кезекті мүшесі 0.0001 мәнінен кіші болғанға дейін есептеуді жалғастыру керек.

4.5.13 Диапазоны минус 30-дан 30-ға дейін болатын, кездейсоқ 20 бүтін сандардан тұратын массивті құру керек. Массивті басып шығару керек. Массивтің максималды және минималды элементтердің орындарын ауыстыру керек. Жаңа массивті экранға шығарыңыз. Әрбір әрекетті әдіс-процедура түрінде көрсетіңіз.

4.5.14 60 нүктенің X және Y координаттары кездейсоқ түрде құрылады. Координаттар мәндерінің диапазоны -150-ден +150-ге дейін. Әр ширекте

орналасқан аралары максимал болатын нүктелердің тізімін экран мониторуна шығару керек. Әдіс-процедураны қолдану керек.

4.5.15 Диапазоны 0-ден9-ға дейін болатын, кездейсоқ 100бүтін сандардан тұратын массивті құру керек. Массивті басып шығару керек. Массивте ең жиі кездесетін санды тауып, экран мониторуна шығару керек. Әрбір әрекетті әдіс-процедура түрінде көрсетіңіз.

4.5.16 Диалог режимінде берілген  $|x| < 1$  және  $m = -1$  үшін әдіс-функцияны қолдана отырып, қатардың соммасын табыңыз. Қатардың кезекті мүшесі 0.0001мәнінен кіші болғанға дейін есептеуді жалғастыру керек:

$$(1+x)^m = 1 + \frac{m}{1!}x + \frac{m(m-1)}{2!}x^2 + \frac{m(m-1)(m-2)}{3!}x^3 + \dots + \frac{m(m-1)\dots(m-n+1)}{n!}x^n + \dots$$

4.5.17 Диапазоны минус 40-тан 60-ке дейін болатын, кездейсоқ 25 бүтін сандардан тұратын массивті құру керек. Массивті басып шығару керек. 3 тах санды тауып, оны массивтің басына орналастырыңыз (сандарға сұрыптауды жүргізуге болмайды). Жаңа массивті экранға шығарыңыз. Әрбір қадамды әдіс-процедура түрінде рәсімдеңіз.

4.5.18 60 нүктенің X және Y координаттары кездейсоқ түрде құрылады. Координаттар мәндерінің диапазоны -150-ден +150-ге дейін. Бір ширекте орналасқан аралары минималды болатын нүктелердің тізімін экран мониторуна шығару керек. Әдіс-процедураны қолдану керек.

4.5.19 Берілген x үшін қатардың соммасын есепте. Қатардың кезекті мүшесі 0.001 мәнінен кіші болғанға дейін есептеуді жалғастыру керек:

$$\arctg(x) = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \dots$$

Әдіс-процедураны қолдану керек.

4.5.20 Жүз тіктөртбұрыштардың қарама-қарсы төбелері A(X,Y) және B(X,Y) кездесок түрде құрылады. Координаттар мәндерінің диапазоны -150-ден +150-ге дейін. Әрбір ширекте орналасқан тіктөртбұрыштардың санын тауып, экран мониторуна шығару керек (егер төбелері әртүрлі ширекте жатса, онда ол тіктөртбұрыш қарастырылмайды). Әдіс-процедураны қолдану керек.

## 4.6 СРСП-да есеп-хатты қорғауға арналған бақылау сұрақтары

4.6.1 C# тілінің класс әдісі түсінігі. Мысал.

4.6.2 C# тілінің класс әдісін жазу форматы.

4.6.3 Класс әдісіне қолжеткізудің қандай спецификаторын білесіз? Мысал.

4.6.4 C# тілінің класстар әдісінің `staticvoidMain()`-мен мәлімет алмасу механизмі?

4.6.5 Класс әдістерінің қандай кіріс параметрлерін білесіз? Мысал.

4.6.6 Класс әдістерінің қандай шығыс параметрлерін білесіз? Мысал.

4.6.7 Класс әдістерінің қандай жағартылған параметрлерін білесіз? Мысал.

4.6.8 C# тіліндегі функция түсінігі. Мысал.

4.6.9 C# тілінде қайда және қашан функция типі жазылады? Мысал.

4.6.10 C# тілінде процедура түсінігі. Мысал.

4.6.11 C# тілінің класс әдістері айнымалыларының локальді және глобальді түсінігі. Мысал.

4.6.12 C# тілінің класс әдістері денесінде қандай «аты» жазу операторы қолданылуы мүмкін?

4.6.13 Бір әдіс ішіне (мысалы, `voidMain()`) басқа әдісті жариялауға бола ма? Мысал.

4.6.14 Рекурсия түсінігі. Мысал.

4.6.15 Рекурсия жетістігі және кемшілігі.

## 5 C# ТІЛІНІҢ КӨПӨЛШЕМДІ МАССИВИ

### 5.1 Бесінші жұмыс мақсаты

Көпөлшемді массивтерді жариялау ережелерін оқу, көпөлшемді матрицаларды қолданып, консоль қосымша құруға практикалық дағдылану.

### 5.2 Теориялық мәліметтер

#### 5.2.1 Көпөлшемді массив түсінігі

Массивтерді бірөлшемді және көпөлшемді бөлу тарихи сипаттама болып табылады. Олардың арасында принципті айырмашылық жоқ. Бір өлшемді массивтер - көпөлшемді массивтердің арнайы жағдайы.

Бірөлшемді массивтер математикалық құрылымды көрсетуге мүмкіндік береді, оларға векторлар, екіөлшемді матрицалар, үшөлшемді - трехмерные – деректер текшелері, жоғары өлшемді массивтер – көпөлшемді деректер текшелері.

Массив өлшемі типтің сипаттамасы болып табылады. Көпөлшемді массивті мәлімдемені жазу форматы бойынша жариялау төмендегідей:

<тип>[, ... ,] <объявители>;

Мысалы: `int[,] matri; int[,] kubi;`

Массив өлшемі жақша ішінде үтір арқылы көрсетіледі.

Үтірлер саны 1-ге артық болса массив өлшемін көрсетеді. Бірөлшемді массивтердегі сияқты көпөлшемді массивтерде де жариялау болады. Біз оны граф тарауында қолданатын боламыз. Әдетте , көп өлшемді массивтер баптандыру массивін құру үшін бағдарламалық қамтамасыз жұмыс істейді.

#### 5.2.2 Массивтердің массиві

C# тілінде массивтердің тағы бір түрі массивтердің массиві болып табылады, оларды басқаша «сынған» массивтер деп атайды (`jaggedarrays`).

Массивтердің массивін бірөлшемді массив деп қарастыруға болады, олардың элементі массив болып табылады, олардың элементінің элементі де массив болып табылады, белгілі бір деңгейге дейін жалғаса береді.

Қандай жағдайларда бұл деректер құрылымдарында қажет болуы мүмкін? Бұл массивтер ағаштарды ұсыну үшін пайдаланылуы мүмкін, онда түйіндер балалардың кез келген санын беруі мүмкін. Бұл, мысалы, отбасы ағаш болады.