

Конспект
за третия ден от упражненията по Програмиране (C++)
със специалност Информатика, редовно, I курс
през 2020-2021 учебна година

Кирил Иванов

• **Пример 3.1**

Този пример показва, как след включване `#include <typeinfo>` чрез `typeid(...).name()` може да се получава уникално название на типа на параметъра на `typeid`.

При това, названието на един и същ тип може да се различава при различните транслятори от C++, но е уникално за типа, т. е. – различно е от названията на другите възможни за C++ типове, в рамките на използването на един и същ транслятор. Тази уникалност е изискване на стандарта на C++.

Стойността, давана от `typeid`, е описание на типа (във вид на съставна данна от вида, наричан структура), който съответният аргумент има по време на изпълнение на кода, а от това описание се извлича названието на типа.

От примера се вижда, че указателят към цял масив е асоцииран с броя на елементите в масива, а указателят към указател не е обвързан с брой на елементи в какъвто и да било масив.

По тази причина е допустимо `for(long elm : *ua)` (ред 41 в кода), където `ua` е указател към масив, но **не е допустимо** `for(long elm : *uu)`, където `uu` е указател към указател (ред 57 в кода).

Примерът се намира във файла: `progr21_03_01_demo.cpp`

• **Задача 3.2**

Да се създаде приложение, което въвежда масив от 6 цели числа и извежда на отделни редове на екрана елементите, които имат един и същ остатък, при прилагане на оператор `%` при делене с три.

Да се напишат две решения на задачата, едното, с достъп до елементите без използване на унарния оператор `*` (имащ операнд указател), и друго, без употреба на оператора `[]` извън декларация. И в двата варианта може да има цикъл от вида `for(... : ...)`.

Примерни решения на задачата съдържат съответно файловете:

`progr21_03_02_index.cpp;`
`progr21_03_02_pointer.cpp.`

• Задача 3.3

Да се създаде приложение, което въвежда масив от 7 числа и след това извежда на отделни редове всички двойки съседни елементи, които имат целочислена сума, заедно с техните индекси.

Да се напишат две решения на задачата, едното, с достъп до елементите без използване на унарния оператор `*` (имащ операнд указател), и друго, без употреба на оператора `[]` извън декларация. И в двата варианта може да има цикъл от вида `for(... .. : ...)`.

Примерни решения на задачата съдържат съответно файловете:

`progr21_03_03_index.cpp;`
`progr21_03_03_pointer.cpp.`

• Задача 3.4

Да се създаде приложение, което:

а) записва в масив с 15 елемента случайни цели числа от 5 до 10;

б) извежда без повторения (по един път) всички числа, срещащи се в масива, и за всяко от тях – на колко места в масива се среща;

в) извежда масива (след записването на случайните стойности).

Да се напишат две решения на задачата, едното, с достъп до елементите без използване на унарния оператор `*` (имащ операнд указател), и друго, без употреба на оператора `[]` извън декларация. И в двата варианта може да има цикъл от вида `for(... .. : ...)`.

Примерни решения на задачата съдържат съответно файловете:

`progr21_03_04_index.cpp;`
`progr21_03_04_pointer.cpp.`

• Задача 3.5

Да се създаде приложение, което:

а) записва в двумерен масив (таблица) от цели числа с 5 реда и 10 колони случайни стойности от 3 до 11 и извежда табулирано масива;

б) пренарежда всяка колона в масива (таблицата) по такъв начин, че максималните за колоната стойности на елементи да попаднат в началото на колоната, а останалите стойности на елементи да запазят наредбата помежду си, и наново извежда табулирано масива.

Да се напишат две решения на задачата, едното, с достъп до елементите без използване на унарния оператор `*` (имащ операнд указател), и друго, без употреба на оператора `[]` извън декларация. И в двата варианта може да има цикъл от вида `for(... .. : ...)`.

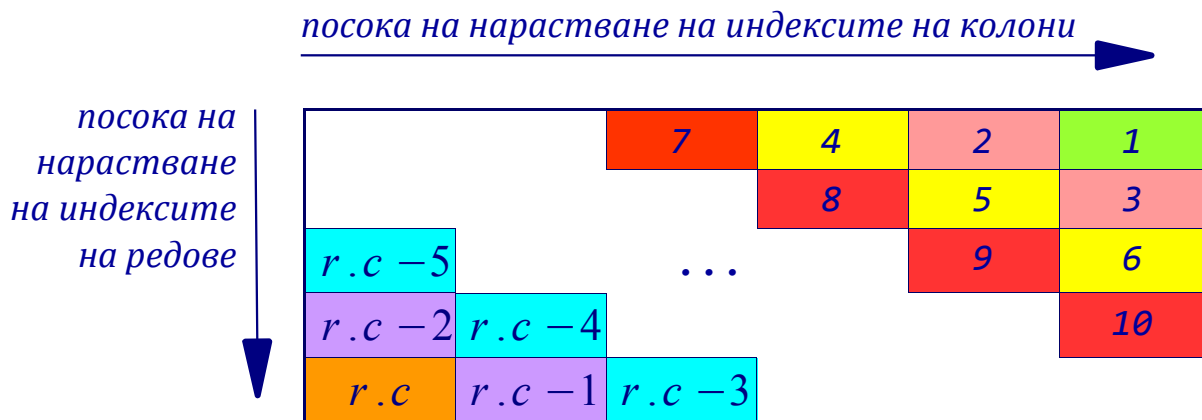
Примерни решения на задачата съдържат съответно файловете:

`progr21_03_05_index.cpp;`
`progr21_03_05_pointer.cpp.`

• Задача 3.6

Да се създаде приложение, което записва в двумерен масив от цели числа с редове и колони, чийто брой се задава с константи, стойности, както показва схемата по-надолу, и извежда табулирано масива.

Схема за попълване на масив с r реда и c колони:



Да се напишат две решения на задачата, едното, с достъп до елементите без използване на унарния оператор `*` (имащ операнд указател), и друго, без употреба на оператора `[]` извън декларация. И в двата варианта може да има цикъл от вида `for(... .. : ...)`.

Примерни решения на задачата съдържат съответно файловете:

```
progr21_03_06_index.cpp;
progr21_03_06_pointer.cpp.
```

• *Пример 3.7*

Този пример илюстрира създаването и използването на взаимосвързани динамични данни.

Той решава задачата:

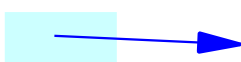
Да се създаде приложение, в което:

а) да са дефинирани данните от приведената по-надолу схема, така че да бъде забранена промяната и на числата, и на указателите (т. е. да бъдат константи);

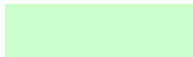
б) стойността, означена с кръг, да се извежда на екрана, като се назовава *само чрез едно име* на данна от схемата. Извеждането да се повтаря чрез използване на всяко от имената поне по два начина – без и с прилагане на оператор [].

Използвани означения:

Указател, като стрелката сочи данната, указвана от указателя:



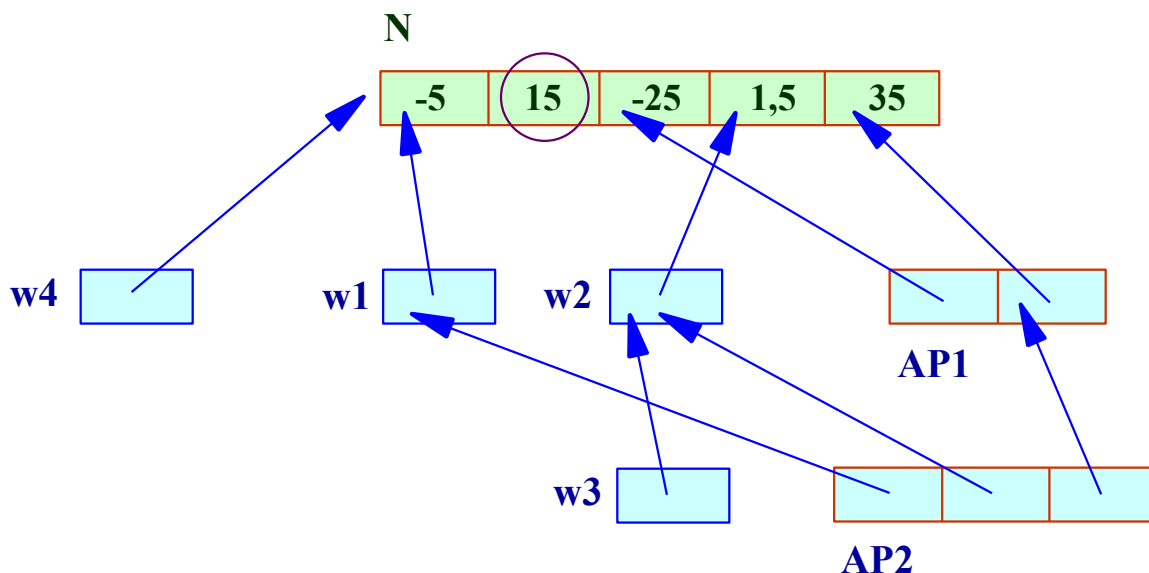
Данна, която не е указател:



Масив:



Схема (**w4** е насочен към целия масив от числа, а всеки друг указател сочи данна, различна от масив):



Примерът се намира във файла: `progr21_03_07_demo.cpp`

• Задача 3.8

Да се създаде приложение, което:

- записва в масив с 15 елемента случайни цели числа от 0 до 200;
- извежда масива на един ред от екрана, а на следващия ред, под всеки елемент извежда индекса на елемента;
- извежда на отделни редове на екрана всички двойки съседни елементи, които имат еднаква четност, и съобщава индексите им.

Да се напишат две решения на задачата, едно с достъп до елементите без унарен оператор *****, прилаган към указател, и друго, без оператор **[]** извън декларация. И в двата варианта може да има цикъл от вида `for(... .. : ...)`.

Примерни решения на задачата съдържат съответно файловете:

`progr21_03_08_index.cpp`;
`progr21_03_08_pointer.cpp`.

• **Задача 3.9**

Да се създаде приложение, което:

а) записва в двумерен масив от цели числа с 5 реда и 8 колони случайни стойности от 5 до 1500 и извежда табулирано масива;

б) табулирано под всяка колона извежда на един ред от екрана сумата от елементите в колоната и на друг ред от екрана минималната стойност на елемент в колоната;

в) извежда сумата на всички елементи в масива и максималния от минимумите по колони.

Да се напишат две решения на задачата, едното, с достъп до елементите без използване на унарния оператор ***** (имащ операнд указател), и друго, без употреба на оператора **[]** извън декларация. И в двата варианта може да има цикъл от вида **for(... .. : ...)**.

Примерни решения на задачата съдържат съответно файловете:

`progr21_03_09_index.cpp;`

`progr21_03_09_pointer.cpp.`