Πανεπιστήμιο Πατρών

Τμήμα Μηχανικών Ηλεκτρονικών Υπολογιστών και Πληροφορικής

ΟΝΤΟΚΕΝΤΡΙΚΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ (C++)

Πίνακες

Περιεχόμενα

- Εισαγωγή
- Πίνακες
- Δήλωση πίνακα
- Παραδείγματα χρήσης πίνακα
- Πέρασμα πίνακα σε συνάρτηση
- Πολυδιάστατοι πίνακες
- Ασκήσεις

Reproduced from the *PowerPoints for C++ How to Program, 4/e by Deitel and Deitel © 2003. Reproduced by permission of Pearson Education, Inc.*

Εισαγωγή

- Πίνακες
  - Δομή δεδομένων για την αποθήκευση ομοειδών στοιχείων
  - Στατική δομή (ίδιο μέγεθος σε όλη τη ζωή του προγράμματος)

Reproduced from the *PowerPoints for C++ How to Program, 4/e by Deitel and Deitel © 2003. Reproduced by permission of Pearson Education, Inc.*

Πίνακες

- Πίνακας
  - Αποθηκεύεται σε συνεχόμενες θέσεις μνήμης
  - Ένα όνομα και ένας τύπος (**`int`**, **`char`**, etc.)
- Προσπέλαση στοιχείων
  - Όνομα πίνακα + θέση στοιχείου
  - Format: ArrayName[PositionNumber]
  - Το πρώτο στοιχείο βρίσκεται στη θέση 0
- Πίνακας c N στοιχείων

  `c[ 0 ]`, **`c[ 1 ]`** … **`c[ n - 1 ]`**
  - Το N στοιχείο βρίσκεται στη θέση N-1

Reproduced from the *PowerPoints for C++ How to Program, 4/e by Deitel and Deitel © 2003. Reproduced by permission of Pearson Education, Inc.*

Πίνακες

- Τα στοιχεία ενός πίνακα συμπεριφέρονται ως μεταβλητές
  - Ανάθεση/Εκτύπωση

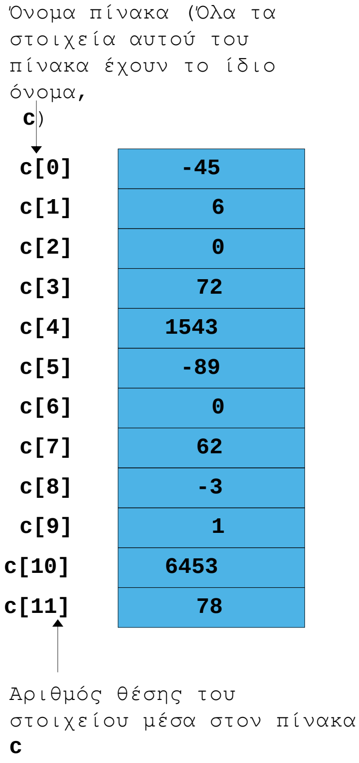
```
c[ 0 ] =  3;
cout << c[ 0 ];
```

- Ο δείκτης ενός πίνακα μπορεί να προκύψει μετά από κάποιο υπολογισμό

  `c[ 5 - 2 ]` ίδιο με **`c[3]`**

Reproduced from the *PowerPoints for C++ How to Program, 4/e by Deitel and Deitel © 2003. Reproduced by permission of Pearson Education, Inc.*

Πίνακες



Όνομα πίνακα (Όλα τα στοιχεία αυτού του πίνακα έχουν το ίδιο όνομα, **c**)

c[0]	-45
c[1]	6
c[2]	0
c[3]	72
c[4]	1543
c[5]	-89
c[6]	0
c[7]	62
c[8]	-3
c[9]	1
c[10]	6453
c[11]	78

Αριθμός θέσης του στοιχείου μέσα στον πίνακα **c**

Reproduced from the *PowerPoints for C++ How to Program, 4/e by Deitel and Deitel © 2003. Reproduced by permission of Pearson Education, Inc.*

Δήλωση πίνακα

- Η δήλωση πίνακα περιλαμβάνει
  - Το όνομα
  - Τον τύπο δεδομένων του πίνακα (των στοιχείων του)
  - Το πλήθος των στοιχείων του
  - *type arrayName*[*arraySize*];

```
int c[ 10 ];  // array of 10 integers
float d[ 3284 ]; // array of 3284 floats
```

- Δήλωση πολλαπλών πινάκων ίδιου τύπου
  - Όπως και οι κανονικές μεταβλητές

```
int b[ 100 ], x[ 27 ];
```

Reproduced from the *PowerPoints for C++ How to Program, 4/e by Deitel and Deitel © 2003. Reproduced by permission of Pearson Education, Inc.*

Παράδειγμα χρήσης πίνακα

- Αρχικοποίηση του πίνακα
  - Με for-loop
    - Θέτει την ίδια τιμή σε κάθε στοιχείο του πίνακα
  - Με λίστα αρχικοποίησης
    - Γίνεται στη δήλωση του πίνακα

```
int n[ 5 ] = { 1, 2, 3, 4, 5 };
```

    - Εάν παραλείψουμε κάποιες τιμές, τα δεξιότερα στοιχεία του πίνακα γίνονται 0
    - Εάν δώσουμε περισσότερες τιμές → syntax error
  - Για να θέσουμε την ίδια τιμή σε κάθε στοιχείο του πίνακα

```
int n[ 5 ] = { 0 };
```

  - Εάν παραλείψουμε το μέγεθος του πίνακα τότε οι τιμές αρχικοποίησης προσδιορίζουν το μέγεθος του πίνακα

```
int n[] = { 1, 2, 3, 4, 5 };
```

Reproduced from the *PowerPoints for C++ How to Program, 4/e by Deitel and Deitel © 2003. Reproduced by permission of Pearson Education, Inc.*

```cpp
// Fig. 4.3: fig04_03.cpp
// Initializing an array.
#include <iostream>

using std::cout;
using std::endl;

#include <iomanip>

using std::setw;

int main()
{
   int n[ 10 ];  // n is an array of

   // initialize elements of array
   for ( int i = 0; i < 10; i++ )
      n[ i ] = 0;   // set element at location i to 0

   cout << "Element" << setw( 13 ) << "Value" << endl;

   // output contents of array n in tabular format
   for ( int j = 0; j < 10; j++ )
      cout << setw( 7 ) << j << setw( 13 ) << n[ j ] << endl;

```

Declare a 10-element array of integers.

Initialize array to **`0`** using a for loop. Note that the array has elements **`n[0]`** to **`n[9]`**.

Outline

**fig04_03.cpp
(1 of 2)**

```cpp
   return 0;  // indicates successful termination

} // end main
```

```
Element        Value
      0            0
      1            0
      2            0
      3            0
      4            0
      5            0
      6            0
      7            0
      8            0
      9            0
```

Outline

**fig04_03.cpp
(2 of 2)**

**fig04_03.cpp
output (1 of 1)**

Outline

fig04_04.cpp
(1 of 1)

```
// Fig. 4.4: fig04_04.cpp
// Initializing an array with a declaration.
#include <iostream>

using std::cout;
using std::endl;

#include <iomanip>

using std::setw;

int main()
{
   // use initializer list to initialize array n
   int n[ 10 ] = { 32, 27, 64, 18, 95, 14, 90, 70, 60, 37 };

   cout << "Element" << setw( 13 ) << "Value" << endl;

   // output contents of array n in tabular format
   for ( int i = 0; i < 10; i++ )
      cout << setw( 7 ) << i << setw( 13 ) << n[ i ] << endl;

   return 0;  // indicates successful termination

} // end main
```

Note the use of the initializer list.

Outline

fig04_04.cpp
output (1 of 1)

```
Element        Value
      0           32
      1           27
      2           64
      3           18
      4           95
      5           14
      6           90
      7           70
      8           60
      9           37
```

Παράδειγμα χρήσης πίνακα

- Μέγεθος πίνακα
  - Μπορεί να προσδιορισθεί με μια σταθερά μεταβλητή (`const`)
    - `const int size = 20;`
  - Οι τιμές των σταθερών δεν μπορούν ν' αλλάξουν
  - Οι σταθερές πρέπει να αρχικοποιηθούν όταν δηλώνονται
  - Οι σταθερές ονομάζονται επίσης read-only μεταβλητές

Reproduced from the *PowerPoints for C++ How to Program, 4/e by Deitel and Deitel © 2003. Reproduced by permission of Pearson Education, Inc.*

Outline

fig04_05.cpp
(1 of 2)

```
// Fig. 4.5: fig04_05.cpp
// Initialize array s to the even integers from 2 to 20.
#include <iostream>

using std::cout;
using std::endl;

#include <iomanip>

using std::setw;

int main()
{
   // constant variable can be used to
   const int arraySize = 10;

   int s[ arraySize ];  // array s has 10 el

   for ( int i = 0; i < arraySize; i++ )  //
      s[ i ] = 2 + 2 * i;

   cout << "Element" << setw( 13 ) << "Value

```

Note use of **`const`** keyword. Only **`const`** variables can specify array sizes.

The program becomes more scalable when we set the array size using a **`const`** variable. We can change **`arraySize`**, and all the loops will still work (otherwise, we'd have to update every loop in the program).

Reproduced from the *PowerPoints for C++ How to Program, 4/e by Deitel and Deitel © 2003. Reproduced by permission of Pearson Education, Inc.*

fig04_05.cpp
(2 of 2)

fig04_05.cpp
output (1 of 1)

```
   // output contents of array s in tabular format
   for ( int j = 0; j < arraySize; j++ )
      cout << setw( 7 ) << j << setw( 13 ) << s[ j ] << endl;

   return 0;  // indicates successful termination

} // end main
```

```
Element        Value
      0            2
      1            4
      2            6
      3            8
      4           10
      5           12
      6           14
      7           16
      8           18
      9           20
```

fig04_06.cpp
(1 of 1)

fig04_06.cpp
output (1 of 1)

```
// Fig. 4.6: fig04_06.cpp
// Using a properly initialized constant variable.
#include <iostream>

using std::cout;
using std::endl;

int main()
{
   const int x = 7;  // initialized constant variable

   cout << "The value of constant variable x is: "
        << x << endl;

   return 0;  // indicates successful termination

} // end main
```

Proper initialization of `const` variable.

```
The value of constant variable x is: 7
```

Outline

fig04_07.cpp
(1 of 1)

fig04_07.cpp
output (1 of 1)

```
// Fig. 4.7: fig04_07.cpp
// A const object must be initi

int main()
{
   const int x;  // Error: x mu

   x = 7;        // Error: cannot modify a const variable

   return 0;     // indicates successful termination

} // end main
```

Uninitialized **`const`** results in a syntax error. Attempting to modify the **`const`** is another error.

```
d:\cpphtp4_examples\ch04\Fig04_07.cpp(6) : error C2734: 'x' :
   const object must be initialized if not extern
d:\cpphtp4_examples\ch04\Fig04_07.cpp(8) : error C2166:
   l-value specifies const object
```

Reproduced from the *PowerPoints for C++ How to Program, 4/e by Deitel and Deitel © 2003. Reproduced by permission of Pearson Education, Inc.*

Outline

fig04_08.cpp
(1 of 1)

fig04_08.cpp
output (1 of 1)

```
// Fig. 4.8: fig04_08.cpp
// Compute the sum of the elements of the array.
#include <iostream>

using std::cout;
using std::endl;

int main()
{
   const int arraySize = 10;

   int a[ arraySize ] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };

   int total = 0;

   // sum contents of array a
   for ( int i = 0; i < arraySize; i++ )
      total += a[ i ];

   cout << "Total of array element values is " << total << endl;

   return 0;  // indicates successful termination

} // end main
```

```
Total of array element values is 55
```

Reproduced from the *PowerPoints for C++ How to Program, 4/e by Deitel and Deitel © 2003. Reproduced by permission of Pearson Education, Inc.*

Outline

fig04_09.cpp
(1 of 2)

```
// Fig. 4.9: fig04_09.cpp
// Histogram printing program.
#include <iostream>

using std::cout;
using std::endl;

#include <iomanip>

using std::setw;

int main()
{
   const int arraySize = 10;
   int n[ arraySize ] = { 19, 3, 15, 7, 11, 9, 13, 5, 17, 1 };

   cout << "Element" << setw( 13 ) << "Value"
        << setw( 17 ) << "Histogram" << endl;

   // for each element of array n, output a bar in histogram
   for ( int i = 0; i < arraySize; i++ ) {
      cout << setw( 7 ) << i << setw( 13 )
           << n[ i ] << setw( 9 );

      for ( int j = 0; j < n[ i ]; j++ )   // print one bar
         cout << '*';
```

Prints asterisks corresponding to size of array element, `n[i]`.

Reproduced from the *PowerPoints for C++ How to Program, 4/e by Deitel and Deitel © 2003. Reproduced by permission of Pearson Education, Inc.*

Outline

fig04_09.cpp
(2 of 2)

fig04_09.cpp
output (1 of 1)

```

      cout << endl;  // start next line of output

   } // end outer for structure

   return 0;  // indicates successful termination

} // end main
```

```
Element        Value       Histogram
      0           19       *******************
      1            3       ***
      2           15       ***************
      3            7       *******
      4           11       ***********
      5            9       *********
      6           13       *************
      7            5       *****
      8           17       *****************
      9            1       *
```

Reproduced from the *PowerPoints for C++ How to Program, 4/e by Deitel and Deitel © 2003. Reproduced by permission of Pearson Education, Inc.*

Outline

fig04_10.cpp
(1 of 2)

```
// Fig. 4.10: fig04_10.cpp
// Roll a six-sided die 6000 times.
#include <iostream>

using std::cout;
using std::endl;

#include <iomanip>

using std::setw;

#include <cstdlib>
#include <ctime>

int main()
{
   const int arraySize = 7;
   int frequency[ arraySize ] = { 0 };

   srand( time( 0 ) );  // seed random-number 

   // roll die 6000 times
   for ( int roll = 1; roll <= 6000; roll++ )
      ++frequency[ 1 + rand() % 6 ]; // replac
                                     // of Fig
```

Remake of old program to roll dice. An array is used instead of 6 regular variables, and the proper element can be updated easily (without needing a **`switch`**).

This creates a number between 1 and 6, which determines the index of **`frequency[]`** that should be incremented.

Outline

fig04_10.cpp
(2 of 2)

fig04_10.cpp
output (1 of 1)

```

   cout << "Face" << setw( 13 ) << "Frequency" << endl;

   // output frequency elements 1-6 in tabular format
   for ( int face = 1; face < arraySize; face++ )
      cout << setw( 4 ) << face
           << setw( 13 ) << frequency[ face ] << endl;

   return 0;  // indicates successful termination

} // end main
```

```
Face    Frequency
   1         1003
   2         1004
   3          999
   4          980
   5         1013
   6         1001
```

Παράδειγμα χρήσης πίνακα

- Strings
  - Πίνακες χαρακτήρων
  - Όλα τα strings τερματίζουν με το null χαρακτήρα (`'\0'`)
  - Παραδείγματα
    - `char string1[] = "hello";`
      - Το **`Null`** θα προστεθεί στο τέλος
      - **Η μεταβλητή `string1`** έχει 6 στοιχεία
    - `char string1[] = { 'h', 'e', 'l', 'l', 'o', '\0′ };`
  - Η προσπέλαση των στοιχείων γίνεται με τον ίδιο τρόπο

    `String1[ 0 ]` is **`'h'`**

    `string1[ 2 ]` is **`'l'`**

Reproduced from the *PowerPoints for C++ How to Program, 4/e by Deitel and Deitel © 2003. Reproduced by permission of Pearson Education, Inc.*

Παράδειγμα χρήσης πίνακα

- Είσοδος από το πληκτρολόγιο

```
char string2[ 10 ];
cin >> string2;
```

  - Αποθηκεύει την είσοδο του χρήστη στο string
    - Μέχρι να συναντήσει τον πρώτο whitespace character
    - Προσθέτει το **`null`** character στο τέλος
  - Εάν το κείμενο υπερβαίνει το μέγεθος του πίνακα →πιθανό runtime error
  - Θέλουμε να το αποφύγουμε αυτό
    - `cin >> setw(10) >> string2;`
    - Διαβάζει 9 χαρακτήρες (χώρος δεσμεύεται και για το '\0')
- Εκτύπωση ενός string
  - `cout << string2 << endl;`
    - Δεν δουλεύει για array άλλων τύπων δεδομένων

Reproduced from the *PowerPoints for C++ How to Program, 4/e by Deitel and Deitel © 2003. Reproduced by permission of Pearson Education, Inc.*

Outline

fig04_12.cpp
(1 of 2)

```
// Fig. 4_12: fig04_12.cpp
// Treating character arrays as strings.
#include <iostream>

using std::cout;
using std::cin;
using std::endl;

int main()
{
   char string1[ 20 ];              //
   char string2[] = "string literal"; /

   // read string from user into ar
   cout << "Enter the string \"hell
   cin >> string1;  // reads "hello" [space terminates input]

   // output strings
   cout << "string1 is: " << string1
        << "\nstring2 is: " << string2;

   cout << "\nstring1 with spaces between characters is:\n";

```

Two different ways to declare strings. **`string2`** is initialized, and its size determined automatically .

Examples of reading strings from the keyboard and printing them out.

Reproduced from the *PowerPoints for C++ How to Program, 4/e by Deitel and Deitel © 2003. Reproduced by permission of Pearson Education, Inc.*

Outline

fig04_12.cpp
(2 of 2)

fig04_12.cpp
output (1 of 1)

```
   // output characters until null character is reached
   for ( int i = 0; string1[ i ] != '\0'; i++ )
      cout << string1[ i ] << ' ';

   cin >> string1;  // reads "there"
   cout << "\nstring1 is: " << string1 << enc

   return 0;  // indicates successful termina

} // end main
```

Can access the characters in a string using array notation. The loop ends when the **`null`** character is found.

```
Enter the string "hello there": hello there
string1 is: hello
string2 is: string literal
string1 with spaces between characters is:
h e l l o
string1 is: there
```

Reproduced from the *PowerPoints for C++ How to Program, 4/e by Deitel and Deitel © 2003. Reproduced by permission of Pearson Education, Inc.*

Παράδειγμα χρήσης πίνακα

- Στατικοί τοπικοί πίνακες
  - Διατηρούν τις τιμές τους μεταξύ των κλήσεων μιας συνάρτησης
  - Οι στατικοί πίνακες αρχικοποιούνται με μηδέν

    ```
    static int array[3];
    ```
- Αυτόματοι τοπικοί πίνακες
  - Δημιουργούνται (και καταστρέφονται) με κάθε κλήση της συνάρτησης.

Reproduced from the *PowerPoints for C++ How to Program, 4/e by Deitel and Deitel © 2003. Reproduced by permission of Pearson Education, Inc.*

Outline

fig04_13.cpp
(1 of 3)

```
// Fig. 4.13: fig04_13.cpp
// Static arrays are initialized to zero.
#include <iostream>

using std::cout;
using std::endl;

void staticArrayInit( void );      // function prototype
void automaticArrayInit( void );   // function prototype

int main()
{
   cout << "First call to each function:\n";
   staticArrayInit();
   automaticArrayInit();

   cout << "\n\nSecond call to each function:\n";
   staticArrayInit();
   automaticArrayInit();
   cout << endl;

   return 0;  // indicates successful termination

} // end main

```

Reproduced from the *PowerPoints for C++ How to Program, 4/e by Deitel and Deitel © 2003. Reproduced by permission of Pearson Education, Inc.*

fig04_13.cpp
(2 of 3)

```
// function to demonstrate a static
void staticArrayInit( void )
{
   // initializes elements to 0 first time function is called
   static int array1[ 3 ];

   cout << "\nValues on entering staticArrayInit:\n";

   // output contents of array1
   for ( int i = 0; i < 3; i++ )
      cout << "array1[" << i << "] = " << array1[ i ] << "  ";

   cout << "\nValues on exiting stati

   // modify and output contents of array1
   for ( int j = 0; j < 3; j++ )
      cout << "array1[" << j << "] = "
           << ( array1[ j ] += 5 ) << "  ";

} // end function staticArrayInit

```

Static array, initialized to zero on first function call.

Array data is changed; the modified values stay.

fig04_13.cpp
(3 of 3)

```
// function to demonstrate an automatic local array
void automaticArrayInit( void )
{
   // initializes elements each time function is called
   int array2[ 3 ] = { 1, 2, 3 };

   cout << "\n\nValues on entering automaticArrayInit:\n";

   // output contents of array2
   for ( int i = 0; i < 3; i++ )
      cout << "array2[" << i << "] = " << array2[ i ] << "  ";

   cout << "\nValues on exiting automaticAr

   // modify and output contents of array2
   for ( int j = 0; j < 3; j++ )
      cout << "array2[" << j << "] = "
           << ( array2[ j ] += 5 ) << "  ";

} // end function automaticArrayInit
```

Automatic array, recreated with every function call.

Although the array is changed, it will be destroyed when the function exits and the changes will be lost.

```
First call to each function:

Values on entering staticArrayInit:
array1[0] = 0  array1[1] = 0  array1[2] = 0
Values on exiting staticArrayInit:
array1[0] = 5  array1[1] = 5  array1[2] = 5

Values on entering automaticArrayInit:
array2[0] = 1  array2[1] = 2  array2[2] = 3
Values on exiting automaticArrayInit:
array2[0] = 6  array2[1] = 7  array2[2] = 8

Second call to each function:

Values on entering staticArrayInit:
array1[0] = 5  array1[1] = 5  array1[2] = 5
Values on exiting staticArrayInit:
array1[0] = 10  array1[1] = 10  array1[2] = 10

Values on entering automaticArrayInit:
array2[0] = 1  array2[1] = 2  array2[2] = 3
Values on exiting automaticArrayInit:
array2[0] = 6  array2[1] = 7  array2[2] = 8
```

fig04_13.cpp
output (1 of 1)

Πέρασμα πίνακα σε συνάρτηση

- Περνάμε το όνομα του πίνακα και προαιρετικά το μέγεθος του πίνακα
  - Παράδειγμα

```
int myArray[ 24 ];
myFunction( myArray, 24 );
```

  - Το μέγεθος του πίνακα είναι καλή πρακτική να περνά ως όρισμα γιατί μας επιτρέπει να έχουμε loops προσπέλασης όλων των στοιχείων εντός της συνάρτησης

Πέρασμα πίνακα σε συνάρτηση

- Το πέρασμα του πίνακα γίνεται με αναφορά (by-reference)
  - Η συνάρτηση μπορεί να αλλάξει τις τιμές των στοιχείων του αρχικού πίνακα
  - Το όνομα του πίνακα αντιστοιχεί στη διεύθυνση του πρώτου στοιχείου του πίνακα
    - Η συνάρτηση γνωρίζει που αποθηκεύεται ο πίνακας στη μνήμη
    - Μπορεί να αλλάξει τις τιμές στις θέσεις μνήμης του πίνακα
- Το πέρασμα μεμονωμένων στοιχείων του πίνακα γίνεται με τιμή (by-value)
  - Όπως συμβαίνει και με τις κανονικές μεταβλητές
  - **`square( myArray[3] );`**

Reproduced from the *PowerPoints for C++ How to Program, 4/e by Deitel and Deitel © 2003. Reproduced by permission of Pearson Education, Inc.*

Πέρασμα πίνακα σε συνάρτηση

- Συναρτήσεις και πίνακες
  - Πρωτότυπο συνάρτησης
    - **`void modifyArray( int b[], int arraySize );`**
    - **`void modifyArray( int [], int );`**
      - Τα ονόματα των παραμέτρων είναι προαιρετικά στο πρωτότυπο
    - Τα παραπάνω πρωτότυπα είναι ισοδύναμα
  - Δεν χρειάζεται το μέγεθος του πίνακα (στο 1ο όρισμα)
    - Αν προσδιορισθεί αγνοείται από τον compiler
  - Εάν δηλώσουμε παράμετρο που είναι πίνακας ως **`const`**
    - Δεν μπορεί να αλλάξει μέσα στη συνάρτηση (compiler error εάν επιχειρηθεί αλλαγή)
    - **`void doNotModify( const int [] );`**

Reproduced from the *PowerPoints for C++ How to Program, 4/e by Deitel and Deitel © 2003. Reproduced by permission of Pearson Education, Inc.*

fig04_14.cpp
(1 of 3)

```
// Fig. 4.14: fig04_14.cpp
// Passing arrays and individual array elements to functions.
#include <iostream>

using std::cout;
using std::endl;

#include <iomanip>

using std::setw;

void modifyArray( int [], int );  // appears strange
void modifyElement( int );

int main()
{
   const int arraySize = 5;              // size of array a
   int a[ arraySize ] = { 0, 1, 2, 3, 4 };  // initialize a

   cout << "Effects of passing entire array by reference:"
        << "\n\nThe values of the original array are:\n";

   // output original array
   for ( int i = 0; i < arraySize; i++ )
      cout << setw( 3 ) << a[ i ];
```

Syntax for accepting an array in parameter list.

Reproduced from the *PowerPoints for C++ How to Program, 4/e by Deitel and Deitel © 2003. Reproduced by permission of Pearson Education, Inc.*

fig04_14.cpp
(2 of 3)

```

   cout << endl;

   // pass array a to modifyArray b
   modifyArray( a, arraySize );

   cout << "The values of the modified array are:\n";

   // output modified array
   for ( int j = 0; j < arraySize; j++ )
      cout << setw( 3 ) << a[ j ];

   // output value of a[ 3 ]
   cout << "\n\n\n"
        << "Effects of passing array e
        << "\n\nThe value of a[3] is "

   // pass array element a[ 3 ] by val
   modifyElement( a[ 3 ] );

   // output value of a[ 3 ]
   cout << "The value of a[3] is " << a[ 3 ] << endl;

   return 0;  // indicates successful termination

} // end main
```

Pass array name (**a**) and size to function. Arrays are passed-by-reference.

Pass a single array element by value; the original cannot be modified.

Reproduced from the *PowerPoints for C++ How to Program, 4/e by Deitel and Deitel © 2003. Reproduced by permission of Pearson Education, Inc.*

Outline

fig04_14.cpp
(3 of 3)

```

// in function modifyArray, "b" points to
// the original array "a" in memory
void modifyArray( int b[], int sizeOfArray )
{
   // multiply each array element by 2
   for ( int k = 0; k < sizeOfArray; k++ )
      b[ k ] *= 2;

} // end function modifyArray

// in function modifyElement, "e" is a loc
// array element a[ 3 ] passed from main
void modifyElement( int e )
{
   // multiply parameter by 2
   cout << "Value in modifyElement is "
        << ( e *= 2 ) << endl;

} // end function modifyElement
```

Although named **b**, the array points to the original array **a**. It can modify **a**'s data.

Individual array elements are passed by value, and the originals cannot be changed.

Reproduced from the *PowerPoints for C++ How to Program, 4/e by Deitel and Deitel © 2003. Reproduced by permission of Pearson Education, Inc.*

Outline

fig04_14.cpp
output (1 of 1)

```
Effects of passing entire array by reference:

The values of the original array are:
  0  1  2  3  4
The values of the modified array are:
  0  2  4  6  8


Effects of passing array element by value:

The value of a[3] is 6
Value in modifyElement is 12
The value of a[3] is 6
```

Reproduced from the *PowerPoints for C++ How to Program, 4/e by Deitel and Deitel © 2003. Reproduced by permission of Pearson Education, Inc.*

Outline

fig04_15.cpp
(1 of 2)

```
1  // Fig. 4.15: fig04_15.cpp
2  // Demonstrating the const type qualifier.
3  #include <iostream>
4
5  using std::cout;
6  using std::endl;
7
8  void tryToModifyArray( const int [] );  //
9
10 int main()
11 {
12    int a[] = { 10, 20, 30 };
13
14    tryToModifyArray( a );
15
16    cout << a[ 0 ] << ' ' << a[ 1 ] << ' ' << a[ 2 ] << '\n';
17
18    return 0;  // indicates successful termination
19
20 } // end main
21
```

Array parameter declared as **`const`**. Array cannot be modified, even though it is passed by reference.

Outline

fig04_15.cpp
(2 of 2)

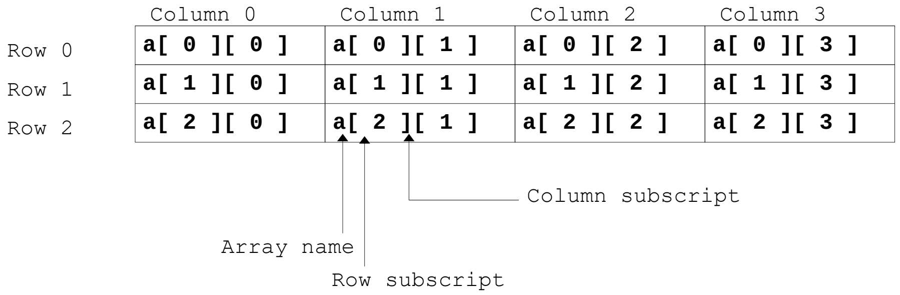
fig04_15.cpp
output (1 of 1)

```
22 // In function tryToModifyArray, "b" cannot be used
23 // to modify the original array "a" in main.
24 void tryToModifyArray( const int b[] )
25 {
26    b[ 0 ] /= 2;    // error
27    b[ 1 ] /= 2;    // error
28    b[ 2 ] /= 2;    // error
29
30 } // end function tryToModifyArray
```

```
d:\cpphtp4_examples\ch04\Fig04_15.cpp(26) : error C2166:
   l-value specifies const object
d:\cpphtp4_examples\ch04\Fig04_15.cpp(27) : error C2166:
   l-value specifies const object
d:\cpphtp4_examples\ch04\Fig04_15.cpp(28) : error C2166:
   l-value specifies const object
```

Πολυδιάστατοι πίνακες

- Δύο διαστάσεις
  - **`a[ i ][ j ]`**
  - Πίνακες με γραμμές και στήλες
  - Πρώτα αναφέρουμε τη γραμμή και μετά τη στήλη
  - “Array of arrays”
    - **`a[0]`** είναι ένας πίνακας 4 στοιχείων
    - **`a[0][0]`** είναι το πρώτο στοιχείο αυτού του πίνακα



	Column 0	Column 1	Column 2	Column 3
Row 0	a[0][0]	a[0][1]	a[0][2]	a[0][3]
Row 1	a[1][0]	a[1][1]	a[1][2]	a[1][3]
Row 2	a[2][0]	a[2][1]	a[2][2]	a[2][3]

Reproduced from the *PowerPoints for C++ How to Program, 4/e by Deitel and Deitel © 2003. Reproduced by permission of Pearson Education, Inc.*

Πολυδιάστατοι πίνακες

- Αρχικοποίηση
  - Default τιμή **`0`**
  - Η λίστα αρχικοποιεί κατά γραμμές

```
int b[ 2 ][ 2 ] = { { 1, 2 }, { 3, 4 } };
```

Row 0 **Row 1**

1	2
3	4

```
int b[ 2 ][ 2 ] = { { 1 }, { 3, 4 } };
```

1	0
3	4

Reproduced from the *PowerPoints for C++ How to Program, 4/e by Deitel and Deitel © 2003. Reproduced by permission of Pearson Education, Inc.*

Πολυδιάστατοι πίνακες

- Προσπέλαση στοιχείου

  `cout << b[ 0 ][ 1 ];`

1	0
3	4

  – Outputs **0**

  – Η παρακάτω αναφορά είναι λανθασμένη

  `cout << b[ 0, 1 ];`

    - Syntax error

- Πρωτότυπα συναρτήσεων

  – Πρέπει να προσδιορίζονται τα μεγέθη των δεικτών

    - Ο πρώτος δείκτης δεν είναι απαραίτητος, όπως και στους μονοδιάστατους πίνακες

  – `void printArray( int [][ 3 ] );`

Outline

fig04_22.cpp
(1 of 2)

```
// Fig. 4.22: fig04_22.cpp
// Initializing multidimensional arrays.
#include <iostream>

using std::cout;
using std::endl;

void printArray( int [][ 3 ] );

int main()
{
   int array1[ 2 ][ 3 ] = { { 1, 2, 3 }, { 4, 5, 6 } };
   int array2[ 2 ][ 3 ] = { 1, 2, 3, 4, 5 };
   int array3[ 2 ][ 3 ] = { { 1, 2 }, { 4 } };

   cout << "Values in array1 by row are:" << endl;
   printArray( array1 );

   cout << "Values in array2 by row are:" << endl;
   printArray( array2 );

   cout << "Values in array3 by row are:" << endl;
   printArray( array3 );

   return 0;  // indicates successful termination

} // end main
```

Note the format of the prototype.

Note the various initialization styles. The elements in **array2** are assigned to the first row and then the second.

Outline

```

// function to output array with two rows
void printArray( int a[][ 3 ] )
{
   for ( int i = 0; i < 2; i++ ) {    // f

      for ( int j = 0; j < 3; j++ )   // output column values
         cout << a[ i ][ j ] << ' ';

      cout << endl;  // start new line of output

   } // end outer for structure

} // end function printArray
```

For loops are often used to iterate through arrays. Nested loops are helpful with multiple-subscripted arrays.

```
Values in array1 by row are:
1 2 3
4 5 6
Values in array2 by row are:
1 2 3
4 5 0
Values in array3 by row are:
1 2 0
4 0 0
```

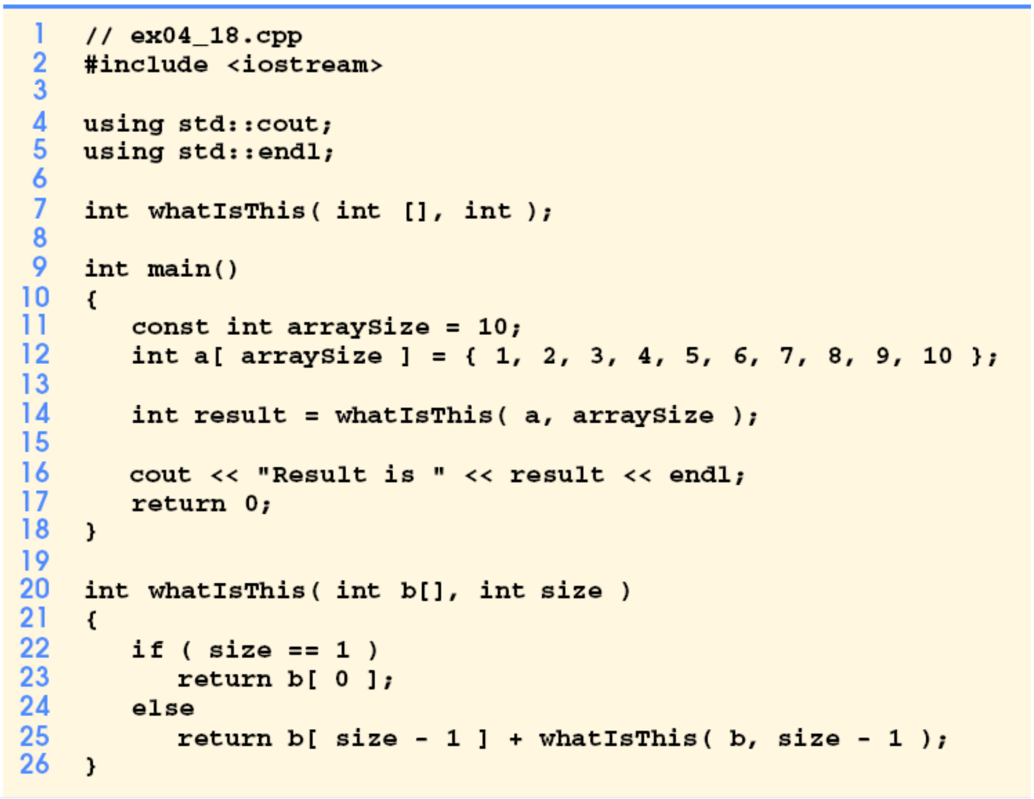
fig04_22.cpp (2 of 2)

fig04_22.cpp output (1 of 1)

Αναφορές

- Harvey M. Deitel, Paul J. Deitel, C++ How to Program, 8th Edition, Prentice Hall.
- Bjarne Stroustrup, The C++ Programming Language, Special Edition, Addison-Wesley.

- Τι κάνει το παρακάτω πρόγραμμα?

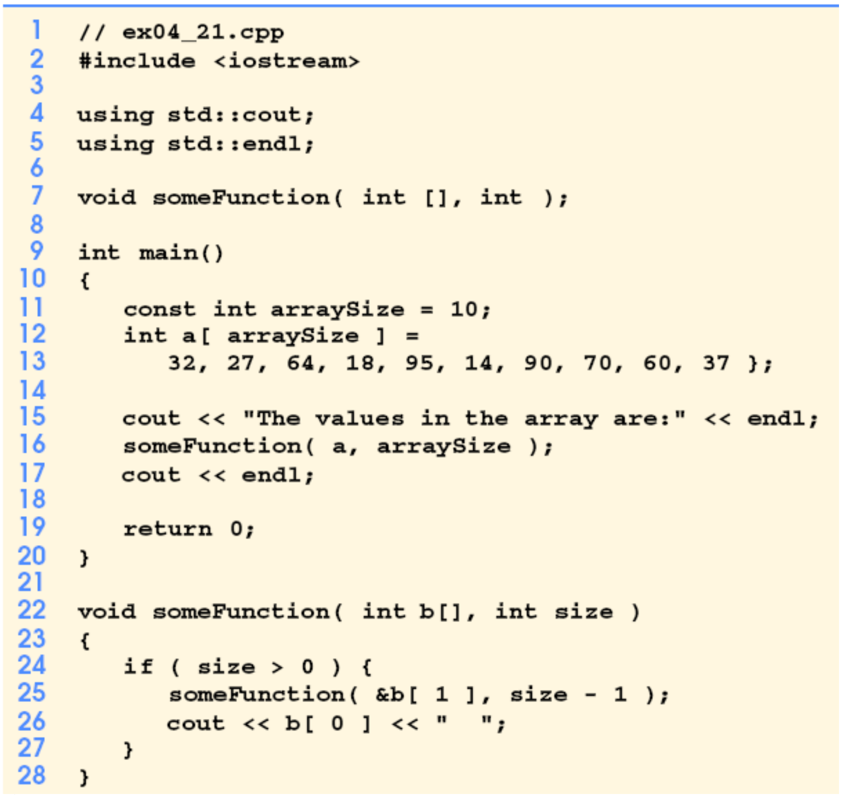


```
// ex04_18.cpp
#include <iostream>

using std::cout;
using std::endl;

int whatIsThis( int [], int );

int main()
{
   const int arraySize = 10;
   int a[ arraySize ] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };

   int result = whatIsThis( a, arraySize );

   cout << "Result is " << result << endl;
   return 0;
}

int whatIsThis( int b[], int size )
{
   if ( size == 1 )
      return b[ 0 ];
   else
      return b[ size - 1 ] + whatIsThis( b, size - 1 );
}
```

Reproduced from the *PowerPoints for C++ How to Program, 4/e by Deitel and Deitel © 2003. Reproduced by permission of Pearson Education, Inc.*

Άσκηση-2

- Τι κάνει το παρακάτω πρόγραμμα?



```
// ex04_21.cpp
#include <iostream>

using std::cout;
using std::endl;

void someFunction( int [], int );

int main()
{
   const int arraySize = 10;
   int a[ arraySize ] =
      32, 27, 64, 18, 95, 14, 90, 70, 60, 37 };

   cout << "The values in the array are:" << endl;
   someFunction( a, arraySize );
   cout << endl;

   return 0;
}

void someFunction( int b[], int size )
{
   if ( size > 0 ) {
      someFunction( &b[ 1 ], size - 1 );
      cout << b[ 0 ] << "  ";
   }
}
```

Reproduced from the *PowerPoints for C++ How to Program, 4/e by Deitel and Deitel © 2003. Reproduced by permission of Pearson Education, Inc.*

Απάντηση-1

- Προσθέτει τα στοιχεία του πίνακα και τυπώνει το άθροισμα (55)

Απάντηση-2

- Τυπώνει τα στοιχεία του πίνακα από το τέλος προς την αρχή.