Διαδικαστικός Προγραμματισμός

Βασίλης Παλιουράς
paliuras@ece.upatras.gr

Άσκηση

- Να γραφεί πρόγραμμα που να αθροίζει δύο διανύσματα Ν στοιχείων σε ISO C90 χρησιμοποιώντας πίνακες ακεραίων.

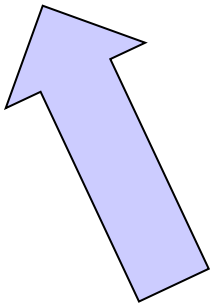
```
void add(const int a[N], const int b[N], int c[N] ) {
        int i;

        for (i=0; i<N; i++)
                c[i] = b[i] + a[i];

        return ;
}

void report (const int c[N]) {
        int i;

        for (i=0; i<N; i++)
                printf("%d ", c[i]);

        printf("\n");
        return ;
}
```

- Πλήρης υλοποίηση συναρτήσεων add, report

Πίνακες ως είσοδοι και έξοδοι

Είσοδοι: δεν επιτρέπεται στη συνάρτηση να αλλάξει τις τιμές στοιχείων πινάκων **const int** []

```
void add(const int a[N], const int b[N], int c[N] ) {
        int i;

        for (i=0; i<N; i++)
                c[i] = b[i] + a[i];

        return ;
}
```



Έξοδος : η συνάρτηση έχει δικαίωμα να Αλλάξει τα στοιχεία του πίνακα **int** []

ΣΥΝΗΘΗ ΛΑΘΗ

```
/* Σωστό !!! */
int main() {
        int a[N] = {1, 2, 3, 4, 5};
        int b[N] = {6, 7, 8, 9, 0};
        int c[N];

        add(a, b, c);
        report (c);

        return 0;
}
```

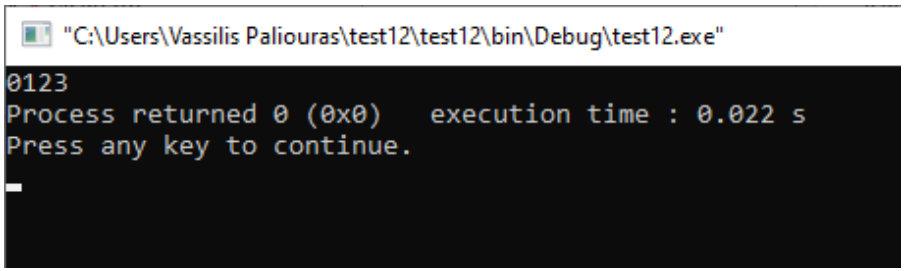
```
c = add(a, b); /* Λάθος: Το c δεν μπορεί να αλλάξει, είναι η
                  διεύθυνση του πρώτου στοιχείου του πίνακα! */

add(a[], b[], c[]); /* Λάθος: Εδώ είναι  syntax error. Μόνο σε
        δήλωση μπορεί να παραληφθεί μια (και μόνο μία) διάσταση (η
        τελευταία). */

add(a[N], b[N], c[N]); /* Λάθος: Η τιμή ενός ακεραίου (έξω από τους
πίνακες) μεταφράζεται σε διεύθυνση!!!  Warning: pointer from integer
without a cast */
```

Buffer overflow

```
#include <stdio.h>
#define N 3

int main ( ) {
int i;

int a[N];

for (i=0; i<N + 1; i++) {
        a[i] = i;
        printf("%d",a[i]);

    }

return 0;
}
```

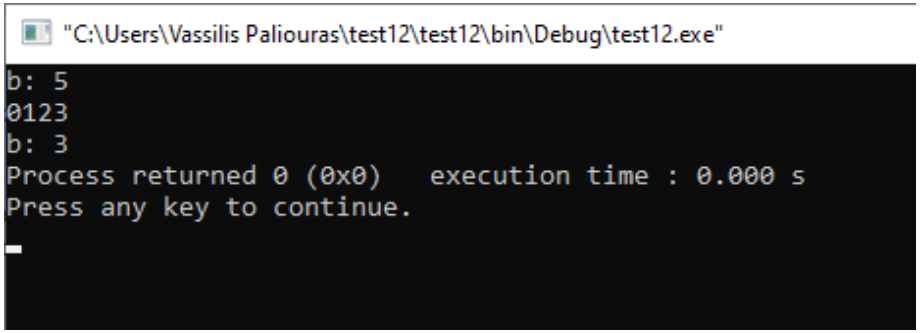
Παραβιάζει το όριο του πίνακα (γράφει σε N+1) στοιχεία

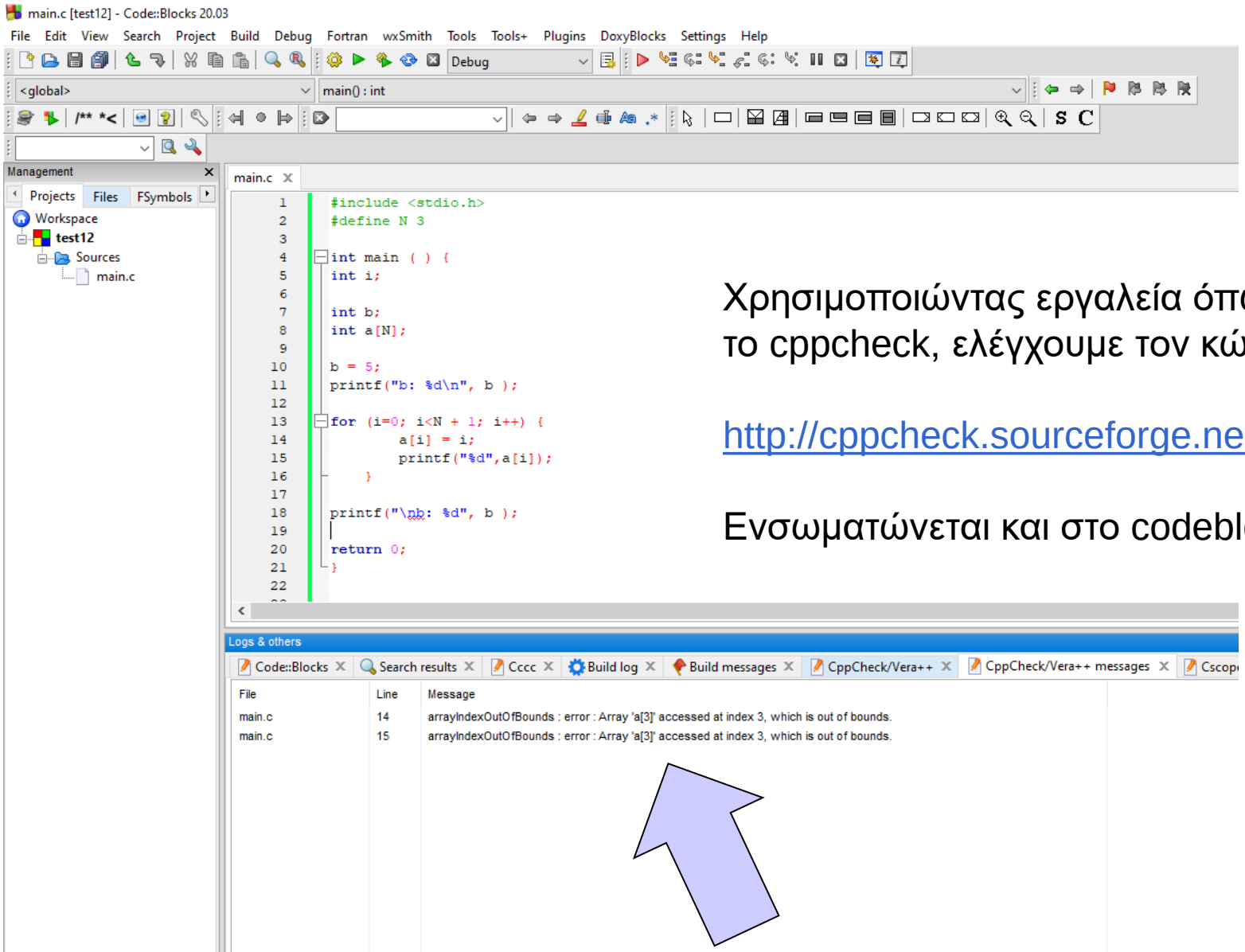
Φαίνεται ότι «δουλεύει»…



```c
#include <stdio.h>
#define N 3

int main ( ) {
int i;

int b;
int a[N];

b = 5;
printf("b: %d\n", b );

for (i=0; i<N + 1; i++) {
        a[i] = i;
        printf("%d",a[i]);
    }

printf("\nb: %d", b );

return 0;
}
```

Αλλά μπορεί να γράψει πάνω σε άλλες μεταβλητές!





Χρησιμοποιώντας εργαλεία όπως το cppcheck, ελέγχουμε τον κώδικα

http://cppcheck.sourceforge.net/

Ενσωματώνεται και στο codeblocks

Το πρόβλημα εντοπίζεται με στατική ανάλυση κώδικα, με cppcheck

Βασικός τύπος **char**

```c
#include <stdio.h>

int main() {

        char a;

        a = 'g';

        printf("this is a char: %c\n", a);

        return 0;
}
```

- Διαφορετική σημασία απλών / διπλών εισαγωγικών στη C
  - Απλά εισαγωγικά => ένας χαρακτήρας
  - Διπλά εισαγωγικά => ακολουθία χαρακτήρων που τερματίζεται με την αξία 0

- Χαρακτήρας και κώδικας ascii

Αλφαριθμητικά (strings)

πρόκειται για **πίνακες χαρακτήρων**:

- `char name[30];`

- αρχικοποίηση με
  `char name[30] = "abcd";`
  το οποίο ισοδυναμεί με

```
name[0] = 'a';
name[1] = 'b';
name[2] = 'c';
name[3] = 'd';
name[4] = 0 ; /* δηλώνει το τέλος ενός
                 αλφαριθμητικού */
```

Ανάγνωση και εκτύπωση αλφαριθμητικού

- `char str[N_MAX];`
- `scanf ("%s", str );`
- `printf ("%s\n", str );`

- `%s` → αντιστοιχεί σε αλφαριθμητικό
- `str[0]` είναι ο πρώτος χαρακτήρας
- `str` είναι η **διεύθυνση του πρώτου στοιχείου**
  - str είναι το **ίδιο** με &str[0]
  - ισχύει για κάθε τύπο πίνακα

Παράδειγμα

- Το σύστημα ζητά από το χρήστη το όνομά του και τυπώνει "hello" ακολουθούμενο από το όνομα του χρήστη.

Υλοποίηση σε C

```
#include <stdio.h>
#include <string.h>
#define N 10

int main ( void  ) {

 char username[N];

 printf("Please enter user name: ");
 scanf("%s", username);

 printf("Hello, %s\n", username);
 printf("Your name is %d letters long", strlen(username));


 return  0;
}
```

το όνομα του πίνακα είναι η διεύθυνση του πρώτου στοιχείου (⇒ δεν βάζουμε &)

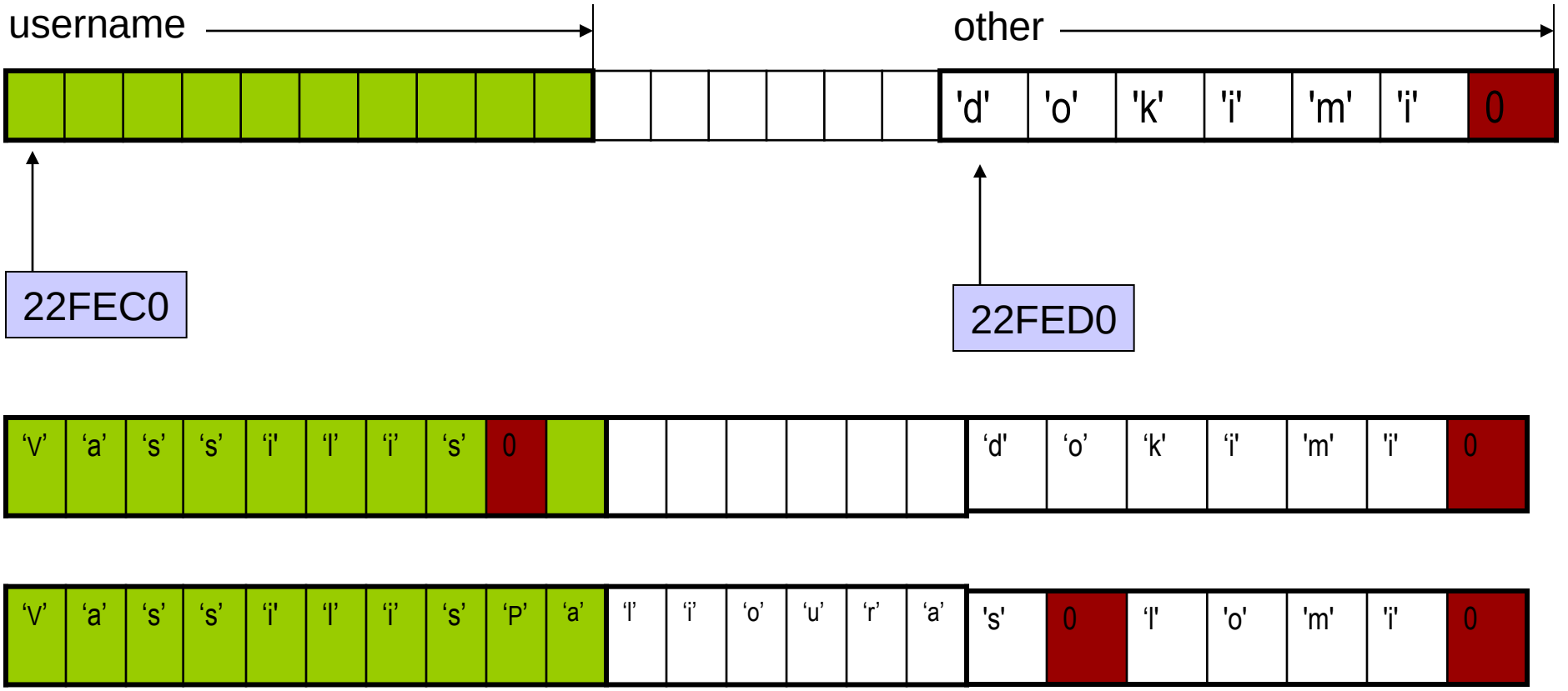
συνάρτηση βιβλιοθήκης strlen()

Υπάρχουν όρια;

```
#include <stdio.h>
#include <string.h>
#define N 10

int main (  ) {

 char other[ ] = "dokimi";
 char username[N];

 printf("Please enter user name: ");
 scanf("%s", username);

 printf("Hello, %s\n", username);
 printf("Your name is %d letters long stored at %X\n", strlen(username), username);

 printf("Value of other: %s at %X", other, other);

 return 0;
}
```

Ναι, αλλά δεν γίνεται έλεγχος...

Σχηματικά η μνήμη – buffer overflow



Καταστρέφονται τα περιεχόμενα της other!

Οι ακριβείς θέσεις των πινάκων μπορεί να είναι διαφορετικές σε διαφορετικά περιβάλλοντα, το πρόβλημα είναι το ίδιο.

```
Please enter user name: VassilisPaliouras
Hello, VassilisPaliouras
Your name is 17 letters long stored at 61FE0F
Value of other: liouras at 61FE19
Process returned 0 (0x0)   execution time : 7.772 s
Press any key to continue.
```

Επεξεργασία ανά χαρακτήρα

```c
#include <stdio.h>
int main()
{
   char string1[ 20 ], string2[] = "string literal";
   int i;

   printf(" Enter a string: ");
   scanf( "%s", string1 );
   printf( "string1 is: %s\nstring2: is %s\n"
           "string1 with spaces between characters is:\n",
           string1, string2 );

   for ( i = 0; string1[ i ] != '\0'; i++ )
      printf( "%c ", string1[ i ] );

   printf( "\n" );
   return 0;
}
```

```
Hello
hello

Process returned 0 (0x0)   execution time : 0.011 s
Press any key to continue.
```

```c
#include <stdio.h>
void DisplayName(char []);

int main ( void ) {
char astring[10] = "Hello";

DisplayName(astring);
DisplayName(astring);

return 0;
}

void DisplayName(char x[]) {
int i;
for (i=0; x[i]!=0 ; i++)
   printf("%c", x[i]);
x[0]='h';
printf("\n");
}
```

```c
#include <stdio.h>
void DisplayName(const char []);

int main ( void ) {
char astring[10] = "Hello";

DisplayName(astring);
DisplayName(astring);

return 0;
}

void DisplayName(const char x[]) {
int i;
for (i=0; x[i]!=0 ; i++)
   printf("%c", x[i]);
x[0]='h';
printf("\n");
}
```

Error: assignment of read-only location

```c
#include <stdio.h>

int main(void) {
   char alphabet[27];    /* 26 letters plus trailing zero */
   char c;

   for (c='A'; c<='Z'; c++)
        alphabet[c-'A'] = c;

   alphabet[c-'A'] = 0;

   printf("%s", alphabet);

   return 0;
}
```

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
Process returned 0 (0x0)   execution time : 0.031 s
Press any key to continue.
```