

ΕΠΛ232 – Προγραμματιστικές Τεχνικές και Εργαλεία

Φροντιστήριο: Τύποι, Πίνακες, Συναρτήσεις
(Κεφάλαια 7-8-9, ΚΝΚ-2ΕΔ)
Τμήμα Πληροφορικής, Πανεπιστήμιο Κύπρου

<http://www.cs.ucy.ac.cy/courses/EPL232>

**Το μάθημα αυτό δομήθηκε βάση των διαλέξεων του
Αναπλ. Καθηγητή Δημήτρη Ζεϊναλιπούρ**



Dr. Andreas Aristidou a.aristidou@ieee.org

Χαρακτήρες (Char)



Dr. Andreas Aristidou a.aristidou@ieee.org

Ανάγνωση και Γράψιμο Χαρακτήρων χρησιμοποιώντας τις **scanf** και **printf**

- Το σύμβολο `%c` επιτρέπει στις `scanf` και `printf` να διαβάζουν και να γράφουν ένα χαρακτήρα:

```
char ch;
```

```
scanf("%c", &ch); /* reads one character */  
printf("%c", ch); /* writes one character */
```

- `scanf` δεν παραλείπει τους χαρακτήρες κενού διαστήματος (space).
- Για να αναγκάσουμε την `scanf` να παραλείπει τους χαρακτήρες κενού διαστήματος γράφουμε ένα space πριν την ανάγνωση του `%c`:

```
scanf(" %c", &ch);
```



University of Cyprus

Dr. Andreas Aristidou

a.aristidou@ieee.org

Ανάγνωση και Γράψιμο Χαρακτήρων χρησιμοποιώντας τις **getchar** και **putchar**

- Για είσοδο και έξοδο μονού χαρακτήρα, οι `getchar` και `putchar` μπορούν να χρησιμοποιηθούν αντί των `scanf` και `printf`.
- Η `putchar` γράφει ένα χαρακτήρα:

```
putchar(ch);
```
- Κάθε φορά που καλείτε η `getchar`, διαβάζει ένα χαρακτήρα:

```
ch = getchar();
```
- Η `getchar` επιστρέφει ένα `int` και όχι μια τιμή `char`, οπότε η `ch` είναι συνήθως τύπου `int`.
- Παρομοίως με την `scanf`, η `getchar` δεν παραλείπει το χαρακτήρα κενού διαστήματος.



University of Cyprus

Dr. Andreas Aristidou

a.aristidou@ieee.org

Ανάγνωση και Γράψιμο Χαρακτήρων χρησιμοποιώντας τις **getchar** και **putchar**

- Η χρήση των `getchar` και `putchar` (αντί των `scanf` και `printf`) εξοικονομεί χρόνο εκτέλεσης.
 - `getchar` και `putchar` είναι πολύ πιο απλές από τις `scanf` και `printf`, που έχουν σχεδιαστεί για να διαβάζουν και να γράφουν πολλά είδη δεδομένων σε μια ποικιλία μορφών.



University of Cyprus

Dr. Andreas Aristidou

a.aristidou@ieee.org

Ανάγνωση και Γράψιμο Χαρακτήρων χρησιμοποιώντας τις **getchar** και **putchar**

- Έστω το `scanf` loop που χρησιμοποιήσαμε για να παραλείψουμε την υπόλοιπη γραμμή εισόδου:

```
do {  
    scanf("%c", &ch);  
} while (ch != '\n');
```

- Ξαναγράφοντας αυτό το loop με χρήση της `getchar`:

```
do {  
    ch = getchar();  
} while (ch != '\n');
```



University of Cyprus

Dr. Andreas Aristidou

a.aristidou@ieee.org

Ανάγνωση και Γράψιμο Χαρακτήρων χρησιμοποιώντας τις `getchar` και `putchar`

- Moving the call of `getchar` into the controlling expression allows us to condense the loop:

```
while ((ch = getchar()) != '\n')  
    ;
```

- The `ch` variable isn't even needed; we can just compare the return value of `getchar` with the new-line character:

```
while (getchar() != '\n')  
    ;
```



University of Cyprus

Dr. Andreas Aristidou

a.aristidou@ieee.org

Παράδειγμα

Προσδιορισμός του μήκους ενός μηνύματος

- Το πρόγραμμα `length.c` τυπώνει το μήκος ενός μηνύματος που δίνεται από το χρήστη:

```
Enter a message: Brevity is the soul of wit.  
Your message was 27 character(s) long.
```

- Το μήκος περιλαμβάνει τα κενά (spaces) και τα σημεία σίξης (punctuation), αλλά όχι τη νέα γραμμή στο τέλος τα πρότασης.
- Μπορούμε να γίνει χρήση τόσο της `scanf` όσο και της `getchar`. Οι περισσότεροι προγραμματιστές σε C θα προτιμούσαν την `getchar`.
- `length2.c` είναι ένα μικρότερο πρόγραμμα που εξαλείφει τη μεταβλητή που χρησιμοποιείται για την αποθήκευση του χαρακτήρα που διαβάζεται από την `getchar`.



University of Cyprus

Dr. Andreas Aristidou

a.aristidou@ieee.org

Παράδειγμα

Προσδιορισμός του μήκους ενός μηνύματος

length.c

```
/* Determines the length of a message */
#include <stdio.h>
int main(void)
{
    char ch;
    int len = 0;
    printf("Enter a message: ");
    ch = getchar();
    while (ch != '\n') {
        len++;
        ch = getchar();
    }
    printf("Your message was %d character(s) long.\n", len);
    return 0;
}
```



University of Cyprus

Dr. Andreas Aristidou

a.aristidou@ieee.org

Παράδειγμα

Προσδιορισμός του μήκους ενός μηνύματος

length2.c

```
/* Determines the length of a message */
#include <stdio.h>
int main(void)
{
    int len = 0;

    printf("Enter a message: ");
    while (getchar() != '\n')
        len++;
    printf("Your message was %d character(s) long.\n", len);

    return 0;
}
```



University of Cyprus

Dr. Andreas Aristidou

a.aristidou@ieee.org

Παράδειγμα

Μετατροπή χαρακτήρων τηλεφώνου σε αριθμό

Γράψτε ένα πρόγραμμα που μετατρέπει τα αλφαβητικά ψηφία ενός τηλεφώνου σε αριθμούς.

Enter phone number: 1-800-COLLECT

Αποτέλεσμα: 1-800-2655328



University of Cyprus

Dr. Andreas Aristidou

a.aristidou@ieee.org

Παράδειγμα

Προσδιορισμός του μήκους ενός μηνύματος

```
#include <stdio.h>
int main(void)
{
    char ch;
    printf("Enter phone number: ");
    while ((ch = getchar()) != '\n')
        switch (ch)
        {
            case 'A': case 'B': case 'C':           putchar('2');           break;
            case 'D': case 'E': case 'F':           putchar('3');           break;
            case 'G': case 'H': case 'I':           putchar('4');           break;
            case 'J': case 'K': case 'L':           putchar('5');           break;
            case 'M': case 'N': case 'O':           putchar('6');           break;
            case 'P': case 'R': case 'S':           putchar('7');           break;
            case 'T': case 'U': case 'V':           putchar('8');           break;
            case 'W': case 'X': case 'Y':           putchar('9');           break;
            default:
                putchar(ch);
                break;
        }
    putchar('\n');
    return 0;
}
```



University of Cyprus

Dr. Andreas Aristidou

a.aristidou@ieee.org

Παράδειγμα Αλλαγή ώρας

Γράψτε ένα πρόγραμμα που παίρνει από τον χρήστη την ώρα σε 12-ώρη μορφή (πμ, μμ) και την μετατρέπει σε 24-ώρη μορφή.

Enter a 12-hours time: 9:11 PM

Equivalent: 21:11



University of Cyprus

Dr. Andreas Aristidou

a.aristidou@ieee.org

Παράδειγμα Αλλαγή ώρας

```
#include <ctype.h>
#include <stdio.h>
int main(void)
{
    int hours12, hours24, minutes;
    char am_pm;

    printf("Enter a 12-hour time: ");
    scanf("%d:%d %c", &hours12, &minutes, &am_pm);

    /* Note: scanf potentially leaves an 'm' in the keyboard buffer. If the
     * program had included additional input, the rest of the input line would
     * need to be skipped before continuing (e.g., using the idiom on p. 141).
     */

    hours24 = hours12 % 12;
    if (toupper(am_pm) == 'P')
        hours24 += 12;

    printf("Equivalent 24-hour time: %.2d:%.2d\n", hours24, minutes);
    return 0;
}
```



University of Cyprus

Dr. Andreas Aristidou

a.aristidou@ieee.org

Arrays



University of Cyprus

Dr. Andreas Aristidou

a.aristidou@ieee.org

Παράδειγμα

Αντιστροφή μιας σειράς αριθμών

- Το πρόγραμμα `reverse.c` ζητά από τον χρήστη να εισάγει μια σειρά από αριθμούς, και στην συνέχεια την εκτυπώνει σε αντίστροφη σειρά:

```
Enter 10 numbers: 34 82 49 102 7 94 23 11 50 31
```

```
In reverse order: 31 50 11 23 94 7 102 49 82 34
```

- Το πρόγραμμα αποθηκεύει τους αριθμούς σε έναν πίνακα καθώς διαβάζονται και, στη συνέχεια, περνά από τον πίνακα ανάποδα, εκτυπώνοντας τα στοιχεία ένα προς ένα.



University of Cyprus

Dr. Andreas Aristidou

a.aristidou@ieee.org

Παράδειγμα Αντιστροφή μιας σειράς αριθμών

reverse.c

```
/* Reverses a series of numbers */  
#include <stdio.h>  
#define N 10  
int main(void)  
{  
    int a[N], i;  
  
    printf("Enter %d numbers: ", N);  
    for (i = 0; i < N; i++)  
        scanf("%d", &a[i]);  
  
    printf("In reverse order:");  
    for (i = N - 1; i >= 0; i--)  
        printf(" %d", a[i]);  
    printf("\n");  
  
    return 0;  
}
```



Παράδειγμα Έλεγχος αριθμού για επαναλαμβανόμενα ψηφία

- Το πρόγραμμα `repdigit.c` ελέγχει αν κάποιο από τα ψηφία ενός αριθμού εμφανίζεται περισσότερες από μία φορές.
- Αφού ο χρήστης εισαγάγει έναν αριθμό, το πρόγραμμα εκτυπώνει είτε `Repeated digit` ή `No repeated digit`:

```
Enter a number: 28212  
Repeated digit
```

- Ο αριθμός 28212 έχει επαναλαμβανόμενο ψηφίο(2), ενώ ο αριθμός 9357 δεν έχει.



Παράδειγμα**Έλεγχος αριθμού για επαναλαμβανόμενα ψηφία**

- Το πρόγραμμα χρησιμοποιεί έναν πίνακα από 10 Boolean τιμές για να παρακολουθεί ποια ψηφία εμφανίζονται σε έναν αριθμό.
- Αρχικά, κάθε στοιχείο της `digit_seen` είναι `false`.
- Όταν δίνεται ένας αριθμός `n`, το πρόγραμμα εξετάζει κατά τα ψηφία του `n` ένα προς ένα, αποθηκεύοντας το τρέχον ψηφίο σε μια μεταβλητή που ονομάζεται `digit`.
 - Εάν το `digit_seen[digit]` είναι `true`, τότε το `digit` εμφανίζεται τουλάχιστον δύο φορές στο `n`.
 - Εάν το `digit_seen[digit]` είναι `false`, τότε το `digit` δεν έχει ξαναεμφανιστεί, οπότε το πρόγραμμα ορίζει το `digit_seen[digit]` σε `true` και συνεχίζει.

**Παράδειγμα****Έλεγχος αριθμού για επαναλαμβανόμενα ψηφία****repdigit.c**

```

/* Checks numbers for repeated digits */
#include <stdbool.h> /* C99 only */
#include <stdio.h>

int main(void)
{
    bool digit_seen[10] = {false};
    int digit;
    long n;

    printf("Enter a number: ");
    scanf("%ld", &n);
    while (n > 0) {
        digit = n % 10;
        if (digit_seen[digit])
            break;
        digit_seen[digit] = true;
        n /= 10;
    }
    if (n > 0)
        printf("Repeated digit\n");
    else
        printf("No repeated digit\n");

    return 0;
}

```



Παράδειγμα Υπολογισμός τόκων

- Το πρόγραμμα `interest.c` Εκτυπώνει έναν πίνακα που δείχνει την αξία των \$100 που επενδύονται με διάφορους συντελεστές τόκων για μια περίοδο ετών.
- Ο χρήστης θα εισαγάγει ένα επιτόκιο και τον αριθμό των ετών που θα επενδυθούν τα χρήματα.
- Ο πίνακας θα δείχνει την αξία των χρημάτων σε διαστήματα ενός έτους — με το επιτόκιο αυτό και τους επόμενους τέσσερις υψηλότερους συντελεστές — υποθέτοντας ότι ο τόκος θα υπολογίζεται μία φορά το χρόνο.



Παράδειγμα Υπολογισμός τόκων

- Δείτε πώς μοιάζει το πρόγραμμα:

Enter interest rate: 6

Enter number of years: 5

Years	6%	7%	8%	9%	10%
1	106.00	107.00	108.00	109.00	110.00
2	112.36	114.49	116.64	118.81	121.00
3	119.10	122.50	125.97	129.50	133.10
4	126.25	131.08	136.05	141.16	146.41
5	133.82	140.26	146.93	153.86	161.05



Παράδειγμα Υπολογισμός τόκων

- Οι αριθμοί στη δεύτερη γραμμή εξαρτώνται από τους αριθμούς στην πρώτη γραμμή, επομένως είναι λογικό να αποθηκεύεται η πρώτη γραμμή σε έναν πίνακα.
 - Οι τιμές στον πίνακα χρησιμοποιούνται στη συνέχεια για τον υπολογισμό της δεύτερης γραμμής.
 - Αυτή η διαδικασία μπορεί να επαναληφθεί για τις επόμενες σειρές.
- Το πρόγραμμα χρησιμοποιεί ένθετες εντολές `for`.
 - Το `outer loop` μετρά από το 1 μέχρι τον αριθμό των ετών που ζήτησε ο χρήστης.
 - Το `inner loop` υπολογίζει το επιτόκιο από τη χαμηλότερη αξία του στην υψηλότερη τιμή του.



Παράδειγμα Υπολογισμός τόκων

interest.c

```

/* Prints a table of compound interest */
#include <stdio.h>

#define NUM_RATES ((int) (sizeof(value) /
    sizeof(value[0])))
#define INITIAL_BALANCE 100.00

int main(void)
{
    int i, low_rate, num_years, year;
    double value[5];

    printf("Enter interest rate: ");
    scanf("%d", &low_rate);
    printf("Enter number of years: ");
    scanf("%d", &num_years);

```

```

    printf("\nYears");
    for (i = 0; i < NUM_RATES; i++) {
        printf("%6d%%", low_rate + i);
        value[i] = INITIAL_BALANCE;
    }
    printf("\n");

    for (year = 1; year <= num_years;
        year++) {
        printf("%3d    ", year);
        for (i = 0; i < NUM_RATES; i++) {
            value[i] += (low_rate + i) /
                100.0 * value[i];
            printf("%7.2f", value[i]);
        }
        printf("\n");
    }

    return 0;
}

```



Multidimensional Arrays



Παράδειγμα

Dealing a Hand of Cards

- Το πρόγραμμα `deal.c` απεικονίζει δύο διαστάσεων πίνακες και σταθερούς πίνακες.
- Το πρόγραμμα μοιράζει ένα τυχαίο χαρτί από μια τυπική τράπουλα.
- Κάθε χαρτί μπορεί να ανήκει σε μια οικογένεια *suit* (clubs, diamonds, hearts, or spades) και έχει ένα βαθμό *rank* (δύο, τρία, τέσσερα, πέντε, έξι, επτά, οχτώ, εννιά, δέκα, βαλές, ντάμα, ρήγας, ή άσσος).

Παράδειγμα

Dealing a Hand of Cards

- Ο χρήστης θα καθορίσει πόσα χαρτιά θα πρέπει να μοιραστούν:

```
Enter number of cards in hand: 5
```

```
Your hand: 7c 2s 5d as 2h
```

- Προβλήματα που πρέπει να επιλυθούν:
 - Πώς θα διαλέξουμε τα τυχαία χαρτιά;
 - Πώς θα αποφύγουμε να πάρουμε το ίδιο φύλλο δύο φορές;



Παράδειγμα

Dealing a Hand of Cards

- Η επιλογή τυχαίων χαρτιών μπορεί αν γίνει με διάφορους τρόπους
 - `time` (from `<time.h>`) – επιστρέφει την τρέχουσα ώρα, κωδικοποιημένη σε έναν μόνο αριθμό.
 - `srand` (from `<stdlib.h>`) – προετοιμάζει τη γεννήτρια τυχαίων αριθμών του C.
 - `rand` (from `<stdlib.h>`) – παράγει έναν φαινομενικά τυχαίο αριθμό κάθε φορά που καλείται.
- Με τη χρήση του `%`, μπορούμε να ορίσουμε την κλίμακα των τιμών που επιστρέφει η `rand` έτσι ώστε να εμπίπτει πάντα μεταξύ του 0 και του 3 (για τα suits) ή μεταξύ του 0 και 12 (για τα ranks).



Παράδειγμα

Dealing a Hand of Cards

- Ο πίνακας `in_hand` χρησιμοποιείται για να παρακολουθεί ποια φύλλα έχουν ήδη επιλεγεί.
- Ο πίνακας έχει 4 γραμμες και 13 στήλες, και κάθε στοιχείο του αντιστοιχεί σε ένα από τα 52 χαρτιά της τράπουλας.
- Όλα τα στοιχεία θα αρχικοποιηθούν σε `false`.
- Κάθε φορά που επιλέγουμε ένα τυχαίο χαρτί, Θα ελέγξουμε αν το στοιχείο της `in_hand` είναι `true` ή `false`.
 - Αν είναι `true`, τότε επιλέγουμε ένα άλλο χαρτί.
 - Αν είναι `false`, τότε αποθηκεύουμε το `true` στο στοιχείο το πίνακα για μελλοντική χρήση.



University of Cyprus

Dr. Andreas Aristidou

a.aristidou@ieee.org

Παράδειγμα

Dealing a Hand of Cards

- Μόλις επαληθεύσουμε ότι ένα χαρτί είναι "νέο", θα πρέπει να μεταφράσουμε την αριθμητική του κατάταξη σε χαρακτήρες και στη συνέχεια να προβάλουμε το χαρτί.
- Για τη μετατροπή του `rank` και του `suit` σε χαρτί τράπουλας, θα ορίσουμε δύο πίνακες χαρακτήρων — μια για το `rank` και μια για το `suit`—και, στη συνέχεια, θα χρησιμοποιήσουμε τους αριθμούς σαν δείκτες στους πίνακες.
- Αυτοί οι πίνακες δεν θα αλλάξουν κατά την εκτέλεση του προγράμματος, οπότε θα οριστούν σαν `const`.



University of Cyprus

Dr. Andreas Aristidou

a.aristidou@ieee.org

Παράδειγμα

Dealing a Hand of Cards

deal.c

```

/* Deals a random hand of cards */
#include <stdbool.h> /* C99 only */
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

#define NUM_SUITS 4
#define NUM_RANKS 13

int main(void)
{
    bool in_hand[NUM_SUITS][NUM_RANKS] = {false};
    int num_cards, rank, suit;
    const char rank_code[] = {'2','3','4','5','6','7','8',
                              '9','t','j','q','k','a'};
    const char suit_code[] = {'c','d','h','s'};

```



Παράδειγμα

Dealing a Hand of Cards

```

srand((unsigned) time(NULL));

printf("Enter number of cards in hand: ");
scanf("%d", &num_cards);

printf("Your hand:");
while (num_cards > 0) {
    suit = rand() % NUM_SUITS; /* picks a random suit */
    rank = rand() % NUM_RANKS; /* picks a random rank */
    if (!in_hand[suit][rank]) {
        in_hand[suit][rank] = true;
        num_cards--;
        printf(" %c%c", rank_code[rank], suit_code[suit]);
    }
}
printf("\n");

return 0;
}

```



Functions



University of Cyprus

Dr. Andreas Aristidou

a.aristidou@ieee.org

Παράδειγμα

Έλεγχος εάν ένας αριθμός είναι πρώτος

- Το πρόγραμμα `prime.c` ελέγχει κατά πόσο ένας αριθμός είναι πρώτος:

```
Enter a number: 34  
Not prime
```

- Το πρόγραμμα χρησιμοποιεί μια συνάρτηση που ονομάζεται `is_prime` που επιστρέφει `true` αν η παράμετρος του είναι πρώτος αριθμός και `false` όταν δεν είναι.
- Η `is_prime` διαιρεί την παράμετρο `n` με καθέναν από τους αριθμούς 2 και την τετραγωνική ρίζα του `n`; Αν το υπόλοιπο είναι ποτέ 0, το `n` δεν είναι πρώτος αριθμός.



University of Cyprus

Dr. Andreas Aristidou

a.aristidou@ieee.org

Παράδειγμα

Έλεγχος εάν ένας αριθμός είναι πρώτος

prime.c

```
/* Tests whether a number is prime */
#include <stdbool.h> /* C99 only */
#include <stdio.h>

bool is_prime(int n)
{
    int divisor;

    if (n <= 1)
        return false;
    for (divisor = 2; divisor * divisor <= n; divisor++)
        if (n % divisor == 0)
            return false;
    return true;
}
```



University of Cyprus

Dr. Andreas Aristidou

a.aristidou@ieee.org

Παράδειγμα

Έλεγχος εάν ένας αριθμός είναι πρώτος

```
int main(void)
{
    int n;

    printf("Enter a number: ");
    scanf("%d", &n);
    if (is_prime(n))
        printf("Prime\n");
    else
        printf("Not prime\n");
    return 0;
}
```



University of Cyprus

Dr. Andreas Aristidou

a.aristidou@ieee.org