



Πανεπιστήμιο Πειραιώς
Τμήμα Ψηφιακών
Συστημάτων

Γλώσσα Προγραμματισμού C
3^η Διάλεξη

Δημοσθένης Κυριαζής

Παρασκευή 19 Οκτωβρίου 2018



Σημερινή διάλεξη

- Σύνοψη 2ης διάλεξης
- Δομές ελέγχου
 - Η εντολή If...Else
 - Η εντολή επανάληψης While
- Διατύπωση αλγορίθμων
 - Επανάληψη ελεγχόμενη από μετρητή
 - Επανάληψη ελεγχόμενη από τιμή φρουρό
 - Ένθετες δομές ελέγχου
 - Σύγκριση “επανάληψης” με μετρητή / τιμή «σημαία»
- Η εντολή επανάληψης for
- Η εντολή πολλαπλής επιλογής switch
- Η εντολή break



Δεσμευμένες λέξεις

□ ANSI C

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

- Περιορισμός για μια γλώσσα, αλλά αυξάνουν την αναγνωσιμότητα και αξιοπιστία των προγραμμάτων και επιταχύνουν τη διαδικασία της μεταγλώττισης
- Πολλές είναι αποδεκτές σε πολλές γλώσσες προγραμματισμού



Μεταβλητές

- Μια μεταβλητή αντιστοιχεί σε μια θέση της μνήμης που έχει το όνομα της μεταβλητής
- Κάθε μεταβλητή
 - όνομα (αναγνωριστικό)
 - τύπος
 - μέγεθος
 - τιμή
- Η δήλωση μιας μεταβλητής προσδιορίζει στο μεταγλωττιστή τον τύπο των δεδομένων που χρησιμοποιεί η μεταβλητή
 - τύπος όνομα-μεταβλητής



Βασικοί τύποι δεδομένων της C

□ Τύπος

χαρακτήρας

ακέραιος

αριθμός κινητής υποδιαστολής

αριθμός κινητής υποδιαστολής, διπλής ακρίβειας

απουσία τιμής

Δεσμευμένη λέξη

char

int

float

double

void

□ Παραδείγματα

```
char ch;
```

```
int counter;
```

```
float f;
```

```
double d;
```

```
float x, y, z;
```



Τελεστές

□ Κατηγορίες τελεστών

Αριθμητικοί

+ - * / %

Σχεσιακοί

> >= == < <= !=

Λογικοί

&& || !

Διαχείρισης μνήμης

& [] . ->

Διαχείρισης των bits ενός byte

>> & | ^ ~



Συναρτήσεις (1/2)

- Σύνολο εντολών που «προσφέρουν» κάποια λειτουργικότητα
- Μορφή συνάρτησης
 - τύπος επιστρεφόμενης-τιμής όνομα συνάρτησης (λίστα παραμέτρων)
 - Η επιστρεφόμενη τιμή αναφέρεται στην έξοδο της συνάρτησης και μπορεί να είναι
 - void: Δεν επιστρέφεται κάτι
 - char, int, float, double: Ένας από τους βασικούς τύπους δεδομένων
 - Η λίστα παραμέτρων αναφέρεται στα ορίσματα / είσοδο της συνάρτησης και μπορεί να είναι
 - void ή (): Δεν λαμβάνει είσοδο
 - Ένας / περισσότεροι από τους βασικούς τύπους με τις σχετικές μεταβλητές
 - Παράδειγμα: `int function1 (int x, char y, int z)`



Συναρτήσεις (2/2)

- Η κλήση των συναρτήσεων γίνεται με το όνομά τους και τα σχετικά ορίσματα

```
int function1 ()
{
    int x;
    x = function2 (3, 4);
    printf("The second function result is %d\n", x);
}
int function2 (int x, int y)
{
    int result;
    result = 0;
    result = x + y;
    return result;
}
```



Γενικές και τοπικές μεταβλητές

εντολές προεπεξεργαστή (#include, #define, ...)

δηλώσεις γενικών μεταβλητών

δηλώσεις συναρτήσεων

```
int main (void)
```

```
{
```

```
εντολές
```

```
}
```

```
τύπος επιστρεφόμενης-τιμής f1(λίστα παραμέτρων)
```

```
{
```

δηλώσεις τοπικών μεταβλητών

```
... }
```

```
:
```

```
τύπος επιστρεφόμενης-τιμής fN (λίστα παραμέτρων)
```

```
{ ... }
```



Δομές ελέγχου

- Σειριακή εκτέλεση
 - Οι εντολές εκτελούνται η μια μετά την άλλη με τη σειρά που είναι γραμμένες
- Μεταφορά ελέγχου
 - Η επόμενη εντολή που εκτελείται δεν είναι η επόμενη στη σειρά
 - Η υπερβολική χρήση εντολών goto οδηγούσε σε πολλά προβλήματα
- Όλα τα προγράμματα γράφονται με βάση 3 δομές ελέγχου
 - Σειριακές δομές
 - Ενσωματωμένες στη C
 - Τα προγράμματα εκτελούνται σειριακά
 - Δομές επιλογής
 - Στη C υπάρχουν τρεις τύποι
 - if, if...else switch
 - Δομές επανάληψης
 - Στη C υπάρχουν τρεις τύποι
 - while, do...while και for



Η εντολή επιλογής `if` (1/4)

□ Εντολή ψευδοκώδικα σε C

```
if ( grade >= 5 )  
    printf( "Passed\n" );
```

- Ο κώδικας C μοιάζει αρκετά με τον ψευδοκώδικα

□ Γενική μορφή εντολής `if`

```
if (έκφραση)  
    εντολή;
```

□ Ρόμβος (σύμβολο απόφασης)

- Υποδεικνύει ότι πρέπει να παρθεί μια απόφαση
- Περιέχει μια έκφραση που μπορεί να είναι `true` (αληθής) ή `false` (ψευδής)
- Έλεγχος της συνθήκης, επιλογή κατάλληλης διαδρομής/ ροής



Η εντολή επιλογής `if...else` (2/4)

□ `if`

- Μια ενέργεια πραγματοποιείται μόνο αν η συνθήκη της `if` είναι `true`

□ `if...else`

- Προσδιορίζει μια ενέργεια που πρέπει να εκτελεστεί τόσο όταν η συνθήκη είναι `true` όσο και όταν η συνθήκη είναι `false`

□ Ψευδοκώδικας:

If ο βαθμός ενός φοιτητή είναι μεγαλύτερος ή ίσος του 5

Τύπωσε "Passed"

else

Τύπωσε "Failed"



Η εντολή επιλογής `if...else` (3/4)

□ Κώδικας C

```
if ( grade >= 5 )  
    printf( "Passed\n");  
else  
    printf( "Failed\n");
```

□ Γενική μορφή εντολής `if...else`

```
if (έκφραση)  
    εντολή1;  
else  
    εντολή2;
```



Η εντολή επιλογής `if...else` (4/4)

□ Τριαδικός τελεστής (`?:`)

■ Έχει τρία ορίσματα

- συνθήκη ? εντολή if true : εντολή if false

□ Παράδειγμα

```
grade >= 5 ? printf("Passed\n") : printf("Failed\n" );
```

```
printf( "%s\n", grade >= 5 ? "Passed" : "Failed" );
```



Σημερινή διάλεξη

- Σύνοψη 2ης διάλεξης
- Δομές ελέγχου
 - Η εντολή If...Else
 - Η εντολή επανάληψης While
- Διατύπωση αλγορίθμων
 - Επανάληψη ελεγχόμενη από μετρητή
 - Επανάληψη ελεγχόμενη από τιμή φρουρό
 - Ένθετες δομές ελέγχου
 - Σύγκριση “επανάληψης” με μετρητή / τιμή «σημαία»
- Η εντολή επανάληψης for
- Η εντολή πολλαπλής επιλογής switch
- Η εντολή break



Η εντολή επιλογής `if...else` (1/5)

- Ένθετες/ «Φωλιασμένες» (nested) εντολές `if...else`
 - Όταν μια εντολή `if` είναι ο προορισμός μιας άλλης εντολής `if` ή `else`, τότε είναι ένθετη μέσα στην πρώτη (εξωτερική) εντολή `if`
 - Παράδειγμα

```
if (count > max) /*εξωτερική if*/  
    if (error) printf("Error, try again."); /*ένθετη if*/
```
 - Στην πράξη σπάνια συναντάται μεγάλο βάθος ένθεσης
 - Ποια `if` σχετίζεται με ποια `else` ;
 - Μια `else` σχετίζεται πάντα με την πλησιέστερη `if` που υπάρχει στο ίδιο τμήμα κώδικα και δεν είναι συσχετισμένη με μια άλλη `if`



Η εντολή επιλογής `if...else` (2/5)

□ Κώδικας C

```
if (grade >= 9)
    printf("A\n");
else
    if (grade >= 8)
        printf("B\n");
    else
        if (grade >= 7)
            printf("C\n");
        else
            if (grade >= 6)
                printf("D\n");
            else
                printf("F\n");
```



Η εντολή επιλογής `if...else` (3/5)

□ Κώδικας C

```
if (grade >= 90)
    printf("A\n");
else if (grade >= 80)
    printf("B\n");
else if (grade >= 70)
    printf("C\n");
else if (grade >= 60)
    printf("D\n");
else
    printf("F\n");
```



Η εντολή επιλογής `if...else` (4/5)

□ Γενική μορφή ένθετων εντολών `if...else`

`if` (έκφραση)

εντολή;

`else if` (έκφραση)

εντολή;

`else if` (έκφραση)

εντολή;

·

·

·

`else`

εντολή;



Η εντολή επιλογής if...else (5/5)

□ Παραδείγματα

```
if (p) /*εξωτερική if*/
    if (q) printf("a and b are true"); /*ένθετη if*/
    else printf("To which statement does this else
apply?");
```

```
if (ch=='s')
{ /*πρώτη if*/
    printf("Enter a number: \n");
    scanf("%d", &y);
    /*δεύτερη if*/
    if (y) printf("Its square is %d.",y*y);
}
    else printf("Make next selection.\n");
```



Σημερινή διάλεξη

- Σύνοψη 2ης διάλεξης
- Δομές ελέγχου
 - Η εντολή If...Else
 - Η εντολή επανάληψης While
- Διατύπωση αλγορίθμων
 - Επανάληψη ελεγχόμενη από μετρητή
 - Επανάληψη ελεγχόμενη από τιμή φρουρό
 - Ένθετες δομές ελέγχου
 - Σύγκριση “επανάληψης” με μετρητή / τιμή «σημαία»
- Η εντολή επανάληψης for
- Η εντολή πολλαπλής επιλογής switch
- Η εντολή break



Η εντολή επανάληψης `while` (1/2)

□ Δομή επανάληψης

- Ο προγραμματιστής ορίζει την επανάληψη μιας ενέργειας για όσο (`while`) κάποια συνθήκη παραμένει αληθής (`true`)

■ Ψευδοκώδικας

While υπάρχουν πράγματα στη λίστα με τα ψώνια

Αγόρασε το επόμενο αντικείμενο και σβήσε το από τη λίστα

■ Βρόχος

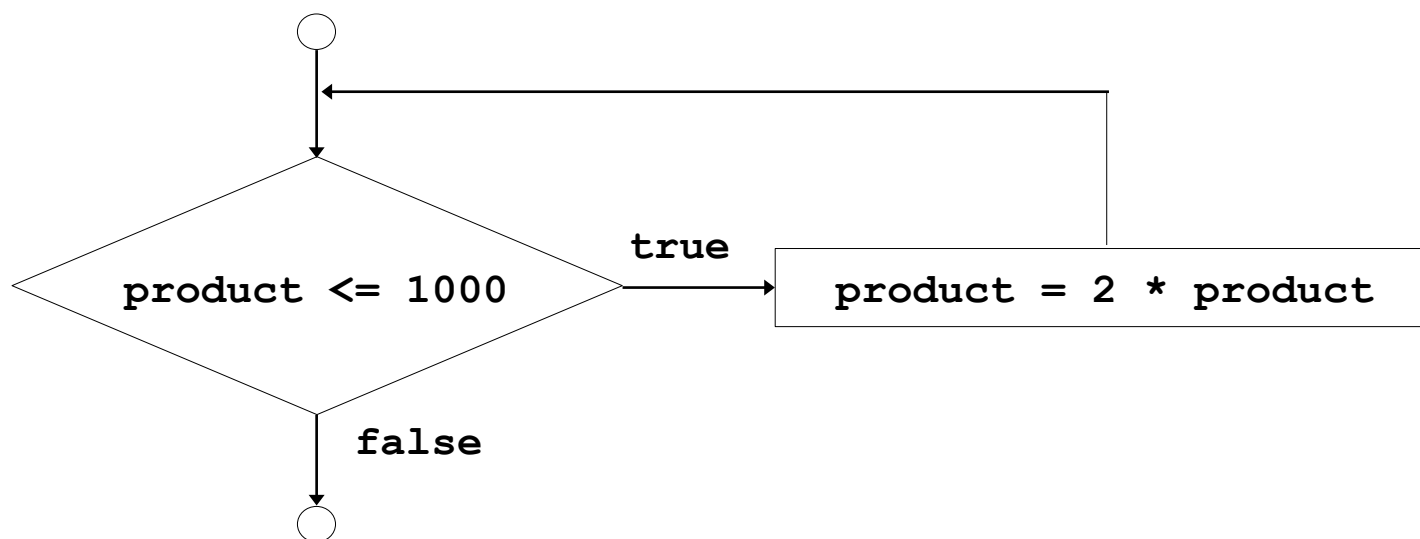
- Ομάδα εντολών που εκτελεί επανειλημμένα ένας υπολογιστής μέχρι κάποια για όσο (`while`) μια συνθήκη παραμένει `true` (αληθής)
- Ο βρόχος `while` επαναλαμβάνεται μέχρι η συνθήκη να γίνει `false`



Η εντολή επανάληψης `while` (2/2)

□ Παράδειγμα

```
int product = 2;  
while ( product <= 1000 )  
    product = 2 * product;
```



Η εντολή επανάληψης do...while (1/2)



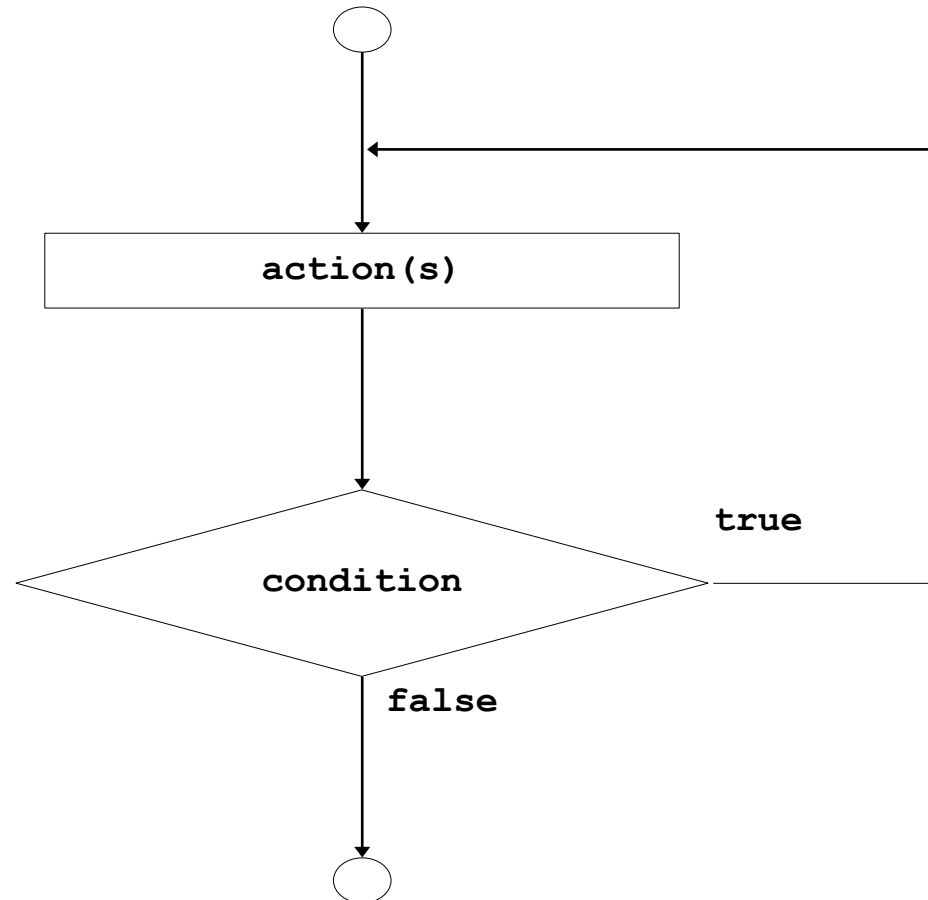
- Παρόμοια με τη while
- Συνθήκη επανάληψης ελέγχεται αφού έχει εκτελεστεί ο κορμός του βρόχου
 - Οι εντολές εκτελούνται τουλάχιστον μια φορά
- Γενική μορφή:

```
do {  
    εντολές;  
} while ( συνθήκη );
```
- Παράδειγμα (έστω counter = 1):

```
do {  
    printf( "%d ", counter );  
} while (++counter <= 10);
```

 - Τυπώνει τους ακέραιους από 1 ως 10

Η εντολή επανάληψης do...while (2/2)





Παράδειγμα

```
1  /* Using the do/while repetition statement */
2  #include <stdio.h>
3
4  /* function main begins program execution */
5  int main()
6  {
7      int counter = 1; /* initialize counter */
8
9      do {
10         printf( "%d ", counter ); /* display counter */
11     } while ( ++counter <= 10 ); /* end do...while */
12
13     return 0; /* indicate program ended successfully */
14
15 } /* end function main */
```

1 2 3 4 5 6 7 8 9 10



Σημερινή διάλεξη

- Σύνοψη 2ης διάλεξης
- Δομές ελέγχου
 - Η εντολή If...Else
 - Η εντολή επανάληψης While
- Διατύπωση αλγορίθμων
 - Επανάληψη ελεγχόμενη από μετρητή
 - Επανάληψη ελεγχόμενη από τιμή φρουρό
 - Ένθετες δομές ελέγχου
 - Σύγκριση “επανάληψης” με μετρητή / τιμή «σημαία»
- Η εντολή επανάληψης for
- Η εντολή πολλαπλής επιλογής switch
- Η εντολή break

Επανάληψη ελεγχόμενη από μετρητή (1/2)



□ Counter-controlled repetition

- Ο βρόχος επαναλαμβάνεται μέχρι ο μετρητής να πάρει μια συγκεκριμένη τιμή
- Συγκεκριμένη επανάληψη: ο αριθμός των επαναλήψεων είναι γνωστός

Επανάληψη ελεγχόμενη από μετρητή (2/2)



- Παράδειγμα 1: Το τμήμα ενός σχολείου αποτελείται από 10 μαθητές. Οι μαθητές έγραψαν διαγώνισμα. Οι βαθμοί είναι ακέραιοι αριθμοί από 0 ως 100. Έστω ότι είναι γνωστοί οι βαθμοί. Γράψτε ένα πρόγραμμα που να διαβάζει τους βαθμούς που του δίνει ο χρήστης και να υπολογίζει το μέσο όρο του τμήματος στο διαγώνισμα.

- Ψευδοκώδικας

Δώσε στο σύνολο την τιμή 0

Δώσε στο μετρητή μαθητών την τιμή 1

While ο μετρητής είναι μικρότερος ή ίσος του 10

Διάβασε τον επόμενο βαθμό

Πρόσθεσε το βαθμό στο σύνολο

Πρόσθεσε ένα στο μετρητή

Δώσε στο μέσο όρο τμήματος την τιμή του πηλίκου του συνόλου διαιρεμένου δια 10

Τύπωσε το μέσο όρο του τμήματος



Πρόγραμμα C (1/2)

```
1  /*Class average program with counter-controlled repetition */
2  #include <stdio.h>
3
4  /* function main begins program execution */
5  int main()
6  {
7      int counter; /* number of grade to be entered next */
8      int grade;   /* grade value */
9      int total;  /* sum of grades input by user */
10     int average; /* average of grades */
11
12     /* initialization phase */
13     total = 0;   /* initialize total */
14     counter = 1; /* initialize loop counter */
15
16     /* processing phase */
17     while ( counter <= 10 ) { /* loop 10 times */
18         printf( "Enter grade: " ); /* prompt for input */
19         scanf( "%d", &grade ); /* read grade from user */
20         total = total + grade; /* add grade to total */
21         counter = counter + 1; /* increment counter */
22     } /* end while */
23
```



Πρόγραμμα C (2/2)

```
24  /* termination phase */
25  average = total / 10;          /* integer division */
26
27  /* display result */
28  printf( "Class average is %d\n", average );
29
30  return 0; /* indicate program ended successfully */
31
32 } /* end function main */
```



Αποτέλεσμα προγράμματος

```
Enter grade: 98
Enter grade: 76
Enter grade: 71
Enter grade: 87
Enter grade: 83
Enter grade: 90
Enter grade: 57
Enter grade: 79
Enter grade: 82
Enter grade: 94
Class average is 81
```



Σημερινή διάλεξη

- Σύνοψη 2ης διάλεξης
- Δομές ελέγχου
 - Η εντολή If...Else
 - Η εντολή επανάληψης While
- Διατύπωση αλγορίθμων
 - Επανάληψη ελεγχόμενη από μετρητή
 - Επανάληψη ελεγχόμενη από τιμή φρουρό
 - Ένθετες δομές ελέγχου
 - Σύγκριση “επανάληψης” με μετρητή / τιμή «σημαία»
- Η εντολή επανάληψης for
- Η εντολή πολλαπλής επιλογής switch
- Η εντολή break

Επανάληψη ελεγχόμενη από τιμή σημαία



- Sentinel-Controlled repetition
- Παράδειγμα 2: Γράψτε ένα πρόβλημα όμοιο με το προηγούμενο με τη διαφορά ότι το πρόγραμμα θα επεξεργάζεται έναν τυχαίο αριθμό βαθμών κάθε φορά που εκτελείται
 - Άγνωστος αριθμός βαθμών
 - Πώς θα ξέρει το πρόγραμμα πότε να σταματήσει;
- Χρήση ειδικής τιμής
 - Τιμή φρουρός (sentinel), σήμα (signal), σημαία (flag)
 - Υποδεικνύει το τέλος της εισαγωγής δεδομένων
 - Ο βρόχος τελειώνει όταν ο χρήστης εισάγει αυτήν την ειδική τιμή
 - Η τιμή αυτή επιλέγεται έτσι ώστε να μην συγχέεται με κάποια κανονική τιμή εισόδου (π.χ. στη συγκεκριμένη περίπτωση μπορεί να είναι -1)

Σχεδίαση αλγόριθμου με τεχνική Top-Down (1/4)



□ Top-down σχεδιασμός

- Διατυπώνουμε σε ψευδοκώδικα τη γενική λειτουργία του προγράμματος (top)

Καθόρισε το μέσο όρο του τμήματος για το διαγώνισμα

- Διαιρούμε τη γενική λειτουργία (top) σε μικρότερες διεργασίες που ακολουθούν μια συγκεκριμένη σειρά

Αρχικοποίησε μεταβλητές

Διάβασε, άθροισε και μέτρησε τους βαθμούς του διαγωνίσματος

Υπολόγισε και τύπωσε το μέσο όρο του τμήματος

□ Στα περισσότερα προγράμματα μπορούμε να διακρίνουμε τρεις φάσεις

- Αρχικοποίηση: αρχικοποιεί τις μεταβλητές του προγράμματος
- Επεξεργασία: παίρνει τις τιμές εισόδου και προσαρμόζει ανάλογα τις μεταβλητές του προβλήματος
- Τερματισμός: υπολογίζει και εκτυπώνει τα τελικά αποτελέσματα

Σχεδίαση αλγόριθμου με τεχνική Top-Down (2/4)



- Αναλύουμε τη φάση αρχικοποίησης από
“*Αρχικοποίησε μεταβλητές*” σε:

Δώσε στο σύνολο την τιμή 0

Δώσε στον μετρητή την τιμή 0

- Αναλύουμε τη φάση “*Διάβασε, άθροισε και μέτρησε τους βαθμούς του διαγωνίσματος*” σε:

Διάβασε τον πρώτο βαθμό (πιθανώς τη «σημαία»)

While ο χρήστης δεν έχει εισάγει τη «σημαία»

Πρόσθεσε το βαθμό στο τρέχον σύνολο

Αύξησε το μετρητή κατά 1

Διάβασε τον επόμενο βαθμό (πιθανώς τη «σημαία»)

Σχεδίαση αλγόριθμου με τεχνική Top-Down (3/4)



- Αναλύουμε τη φάση “Υπολόγισε και εκτύπωσε το μέσο όρο του τμήματος” ως εξής:

If ο μετρητής δεν είναι ίσος με 0

Κάνε το μέσο όρο ίσο με το πηλίκο του συνόλου δια του μετρητή

Τύπωσε το μέσο όρο

else

Τύπωσε “No grades were entered”

Σχεδίαση αλγόριθμου με τεχνική Top-Down (4/4)



Δώσε στο σύνολο την τιμή 0

Δώσε στον μετρητή την τιμή 0

Διάβασε τον πρώτο βαθμό (πιθανώς τη «σημαία»)

While ο χρήστης δεν έχει εισάγει τη «σημαία»

Πρόσθεσε το βαθμό στο τρέχον σύνολο

Αύξησε το μετρητή κατά 1

Διάβασε τον επόμενο βαθμό (πιθανώς τη «σημαία»)

If ο μετρητής δεν είναι ίσος με 0

Κάνε το μέσο όρο ίσο με το πηλίκο του συνόλου δια του μετρητή

Τύπωσε το μέσο όρο

else

Τύπωσε “No grades were entered”



Πρόγραμμα C (1/2)

```
1  /* Class average program with sentinel-controlled repetition */
2  #include <stdio.h>
3
4  /* function main begins program execution */
5  int main()
6  {
7      int counter;    /* number of grades entered */
8      int grade;     /* grade value */
9      int total;     /* sum of grades */
10
11     float average; /* number with decimal point for average */
12
13     /* initialization phase */
14     total = 0;     /* initialize total */
15     counter = 0;   /* initialize loop counter */
16
17     /* processing phase */
18     /* get first grade from user */
19     printf( "Enter grade, -1 to end: " );    /* prompt for input */
20     scanf( "%d", &grade );                 /* read grade from user */
21
22     /* loop while sentinel value not yet read from user */
23     while ( grade != -1 ) {
24         total = total + grade;               /* add grade to total */
25         counter = counter + 1;              /* increment counter */
26     }
```



Πρόγραμμα C (2/2)

```
27     printf( "Enter grade, -1 to end: " ); /* prompt for input */
28     scanf("%d", &grade); /* read next grade */
29 } /* end while */
30
31 /* termination phase */
32 /* if user entered at least one grade */
33 if ( counter != 0 ) {
34
35     /* calculate average of all grades entered */
36     average = ( float ) total / counter;
37
38     /* display average with two digits of precision */
39     printf( "Class average is %.2f\n", average );
40 } /* end if */
41 else { /* if no grades were entered, output message */
42     printf( "No grades were entered\n" );
43 } /* end else */
44
45 return 0; /* indicate program ended successfully */
46
47 } /* end function main */
```



Αποτέλεσμα προγράμματος

```
Enter grade, -1 to end: 75
Enter grade, -1 to end: 94
Enter grade, -1 to end: 97
Enter grade, -1 to end: 88
Enter grade, -1 to end: 70
Enter grade, -1 to end: 64
Enter grade, -1 to end: 83
Enter grade, -1 to end: 89
Enter grade, -1 to end: -1
Class average is 82.50
Enter grade, -1 to end: -1
No grades were entered
```



Παρατηρήσεις

- Καθώς η μεταβλητή counter λαμβάνει μη μηδενικές τιμές (μη προσημασμένος ακέραιος) ορίζεται ως
 - `unsigned int counter;`
- Χρήση τελεστή αλλαγής τύπου (ρητή μετατροπή)
 - π.χ. `average = (float) total / counter;`
 - Έμμεση μετατροπή του τελεστέου total: δημιουργείται αντίγραφο του τελεστέου και μετατρέπεται στο «νέο» τύπο
 - Δέχεται μόνο 1 τελεστέο!
- Μορφοποίηση αριθμών κινητής υποδιαστολής
 - π.χ. `printf ("%.2f\n", 7.777) // 7.78`
 - Χρήση αναγνωριστικού μετατροπής `%f` για τον καθορισμό της ακρίβειας προβολής



Σημερινή διάλεξη

- Σύνοψη 2ης διάλεξης
- Δομές ελέγχου
 - Η εντολή If...Else
 - Η εντολή επανάληψης While
- Διατύπωση αλγορίθμων
 - Επανάληψη ελεγχόμενη από μετρητή
 - Επανάληψη ελεγχόμενη από τιμή φρουρό
 - Ένθετες δομές ελέγχου
 - Σύγκριση “επανάληψης” με μετρητή / τιμή «σημαία»
- Η εντολή επανάληψης for
- Η εντολή πολλαπλής επιλογής switch
- Η εντολή break



Ένθετες δομές ελέγχου (1/6)

- Nested control structures
- Παράδειγμα: Δίνεται η λίστα των αποτελεσμάτων ενός διαγωνίσματος 10 φοιτητών σε ένα κολέγιο. 1 είναι ο κωδικός επιτυχίας ενώ 2 είναι ο κωδικός αποτυχίας. Γράψτε ένα πρόγραμμα που να υπολογίζει και να τυπώνει τον αριθμό των επιτυχόντων και αποτυχόντων. Αν ο αριθμός των επιτυχόντων είναι μεγαλύτερος από 8 το πρόγραμμα θα πρέπει επίσης να τυπώνει ένα μήνυμα.



Ένθετες δομές ελέγχου (2/6)

□ Παρατηρούμε ότι

- Το πρόγραμμα πρέπει να επεξεργαστεί 10 αποτελέσματα διαγωνίσματος
 - Άρα μπορούμε να χρησιμοποιήσουμε επανάληψη ελεγχόμενη από μετρητή
- Θα πρέπει να χρησιμοποιήσουμε δύο μετρητές
 - Αριθμός επιτυχόντων
 - Αριθμός αποτυχόντων
- Κάθε αποτέλεσμα είναι ένας αριθμός: 1 ή 2
 - Αν ο αριθμός δεν είναι 1, μπορούμε να υποθέσουμε ότι είναι 2



Ένθετες δομές ελέγχου (3/6)

- Γενική λειτουργία προγράμματος

Ανάλυσε τα αποτελέσματα της εξέτασης και αποφάσισε αν είναι καλά

- Πρώτη ανάλυση

Αρχικοποίησε μεταβλητές

Διάβασε τους δέκα βαθμούς και υπολόγισε τον αριθμό επιτυχόντων και αποτυχόντων

Τύπωσε μια περίληψη των αποτελεσμάτων της εξέτασης και αποφάσισε αν τα αποτελέσματα είναι καλά

- Ανάλυση της πρότασης “Αρχικοποίησε μεταβλητές” σε:

Αρχικοποίησε επιτυχίες με μηδέν

Αρχικοποίησε αποτυχίες με μηδέν

Αρχικοποίησε μετρητή φοιτητών με ένα



Ένθετες δομές ελέγχου (4/6)

- Ανάλυση της “*Διάβασε τους δέκα βαθμούς και υπολόγισε τον αριθμό επιτυχόντων και αποτυχόντων*” σε

While ο μετρητής φοιτητών είναι μικρότερος ή ίσος του δέκα

Διάβασε το επόμενο αποτέλεσμα

If ο φοιτητής πέρασε

Αύξησε τις επιτυχίες κατά ένα

else

Αύξησε τις αποτυχίες κατά ένα

Αύξησε το μετρητή φοιτητών κατά ένα



Ένθετες δομές ελέγχου (5/6)

- Ανάλυση της “*Τύπωσε μια περίληψη των αποτελεσμάτων και αποφάσισε αν τα αποτελέσματα είναι καλά*” σε
 - Τύπωσε τον αριθμό των επιτυχιών*
 - Τύπωσε τον αριθμό των αποτυχιών*
 - If περισσότεροι από οκτώ φοιτητές πέρασαν*
 - Τύπωσε “Good results!”*



Ένθετες δομές ελέγχου (6/6)

Αρχικοποίησε επιτυχίες με μηδέν

Αρχικοποίησε αποτυχίες με μηδέν

Αρχικοποίησε μετρητή φοιτητών με ένα

While ο μετρητής φοιτητών είναι μικρότερος ή ίσος του δέκα

Διάβασε το επόμενο αποτέλεσμα

If ο φοιτητής πέρασε

 Αύξησε τις επιτυχίες κατά ένα

else

 Αύξησε τις αποτυχίες κατά ένα

Αύξησε το μετρητή φοιτητών κατά ένα

Τύπωσε τον αριθμό των επιτυχιών

Τύπωσε τον αριθμό των αποτυχιών

If περισσότεροι από οκτώ φοιτητές πέρασαν

 Τύπωσε “Good results!”



Πρόγραμμα C (1/2)

```
1  /* Analysis of examination results */
2  #include <stdio.h>
3
4  /* function main begins program execution */
5  int main()
6  {
7      /* initialize variables in definitions */
8      int passes = 0;    /* number of passes */
9      int failures = 0; /* number of failures */
10     int student = 1;  /* student counter */
11     int result;       /* one exam result */
12
13     /* process 10 students using counter-controlled loop */
14     while ( student <= 10 ) {
15
16         /* prompt user for input and obtain value from user */
17         printf( "Enter result ( 1=pass,2=fail ): " );
18         scanf( "%d", &result );
19
20         /* if result 1, increment passes */
21         if ( result == 1 ) {
22             passes = passes + 1;
23         } /* end if */

```



Πρόγραμμα C (2/2)

```
24     else { /* otherwise, increment failures */
25         failures = failures + 1;
26     } /* end else */
27
28     student = student + 1; /* increment student counter */
29 } /* end while */
30
31 /* termination phase; display number of passes and failures */
32 printf( "Passed %d\n", passes );
33 printf( "Failed %d\n", failures );
34
35 /* if more than eight students passed, print "Good results" */
36 if ( passes > 8 ) {
37     printf( "Good Results!\n" );
38 } /* end if */
39
40 return 0; /* indicate program ended successfully */
41
42 } /* end function main */
```



Αποτελέσματα προγράμματος

```
Enter Result (1=pass,2=fail) : 1
Enter Result (1=pass,2=fail) : 2
Enter Result (1=pass,2=fail) : 2
Enter Result (1=pass,2=fail) : 1
Enter Result (1=pass,2=fail) : 1
Enter Result (1=pass,2=fail) : 1
Enter Result (1=pass,2=fail) : 2
Enter Result (1=pass,2=fail) : 1
Enter Result (1=pass,2=fail) : 1
Enter Result (1=pass,2=fail) : 2
Passed 6
Failed 4
```

```
Enter Result (1=pass,2=fail) : 1
Enter Result (1=pass,2=fail) : 1
Enter Result (1=pass,2=fail) : 1
Enter Result (1=pass,2=fail) : 2
Enter Result (1=pass,2=fail) : 1
Enter Result (1=pass,2=fail) : 1
Enter Result (1=pass,2=fail) : 1
Enter Result (1=pass,2=fail) : 1
Enter Result (1=pass,2=fail) : 1
Enter Result (1=pass,2=fail) : 1
Passed 9
Failed 1
Good results!
```



Σημερινή διάλεξη

- Σύνοψη 2ης διάλεξης
- Δομές ελέγχου
 - Η εντολή If...Else
 - Η εντολή επανάληψης While
- Διατύπωση αλγορίθμων
 - Επανάληψη ελεγχόμενη από μετρητή
 - Επανάληψη ελεγχόμενη από τιμή φρουρό
 - Ένθετες δομές ελέγχου
 - Σύγκριση “επανάληψης” με μετρητή / τιμή «σημαία»
- Η εντολή επανάληψης for
- Η εντολή πολλαπλής επιλογής switch
- Η εντολή break

Σύγκριση “επανάληψης” με μετρητή / τιμή «σημαία»



- Επανάληψη ελεγχόμενη από μετρητή
 - Καθορισμένη επανάληψη: ξέρουμε πόσες φορές θα εκτελεστεί ο βρόχος
 - Χρήση μεταβλητής ελέγχου για τη μέτρηση των επαναλήψεων
- Επανάληψη ελεγχόμενη από τιμή “σημαία”
 - Αόριστη επανάληψη
 - Χρησιμοποιείται όταν ο αριθμός των επαναλήψεων δεν είναι γνωστός
 - Η τιμή σημαία σημαίνει το τέλος των δεδομένων

Κύρια χαρακτηριστικά επανάληψης με μετρητή



□ Χρειάζεται

- Μια μεταβλητή ελέγχου (ή μετρητής βρόχου)
- Αρχικοποίηση της μεταβλητής ελέγχου
- Μια αύξηση (ή μείωση) μέσω της οποίας η εντολή ελέγχου μεταβάλλεται κάθε φορά μέσα στο βρόχο
- Μια συνθήκη που ελέγχει την τιμή της μεταβλητής ελέγχου κι ανάλογα να συνεχίζεται ή να τερματίζεται ο βρόχος

Κύρια χαρακτηριστικά επανάληψης με μετρητή



□ Παράδειγμα

```
int counter = 1;           // αρχικοποίηση
while ( counter <= 10 ) { // επανάληψη
    printf( "%d\n", counter );
    ++counter;             // αύξηση
}
```

□ Εναλλακτικά

```
int counter = 1;           // αρχικοποίηση
while ( ++counter <= 10 ) // αύξηση και επανάληψη
{
    printf( "%d\n", counter );
}
```



Πρόγραμμα C

```
1  /* Counter-controlled repetition */
2  #include <stdio.h>
3
4  /* function main begins program execution */
5  int main()
6  {
7      int counter = 1;          /* initialization */
8
9      while ( counter <= 10 ) { /* repetition condition */
10         printf ( "%d\n", counter ); /* display counter */
11         ++counter;             /* increment */
12     } /* end while */
13
14     return 0; /* indicate program ended successfully */
15
16 } /* end function main */
```

```
1
2
3
4
5
6
7
8
9
10
```



Σημερινή διάλεξη

- Σύνοψη 2ης διάλεξης
- Δομές ελέγχου
 - Η εντολή If...Else
 - Η εντολή επανάληψης While
- Διατύπωση αλγορίθμων
 - Επανάληψη ελεγχόμενη από μετρητή
 - Επανάληψη ελεγχόμενη από τιμή φρουρό
 - Ένθετες δομές ελέγχου
 - Σύγκριση “επανάληψης” με μετρητή / τιμή «σημαία»
- Η εντολή επανάληψης for
- Η εντολή πολλαπλής επιλογής switch
- Η εντολή break



Η εντολή επανάληψης for (1/5)

- Χρησιμοποιείται για την επανάληψη μιας εντολής ή μιας ομάδας εντολών κατά έναν προσδιοριζόμενο αριθμό φορών
- Γενική μορφή εντολής for
for (αρχικοποίηση; συνθήκη-ελέγχου; αύξηση/μείωση)
εντολή;
 - Αρχικοποίηση
 - παροχή αρχικής τιμής στη μεταβλητή που ελέγχει το βρόχο
 - εκτελείται μόνο μια φορά πριν ξεκινήσει ο βρόχος
 - Συνθήκη-ελέγχου
 - ελέγχει την τιμή της μεταβλητής ελέγχου και τη συγκρίνει με μια τιμή προορισμού στην αρχή του βρόχου
 - Αν το αποτέλεσμα είναι true ο βρόχος επαναλαμβάνεται
 - Αν το αποτέλεσμα είναι false ο βρόχος σταματά και η εκτέλεση του προγράμματος συνεχίζει με την εντολή που βρίσκεται αμέσως μετά από το βρόχο



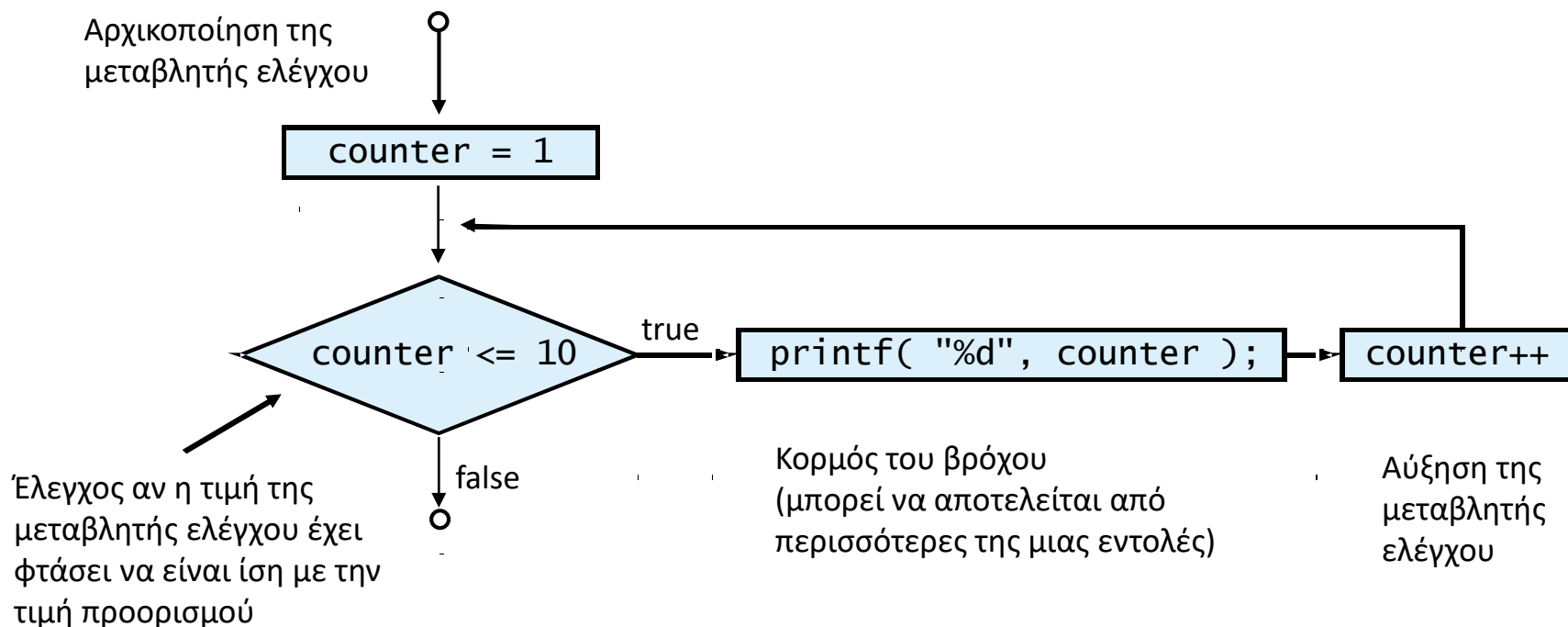
Η εντολή επανάληψης for (2/5)

- Γενική μορφή εντολής for
for (αρχικοποίηση; συνθήκη-ελέγχου; αύξηση/μείωση)
εντολή;
- Αύξηση/ μείωση
 - εκτελείται στο τέλος του βρόχου, αφού ολοκληρωθεί η εκτέλεση της εντολής ή της ομάδας εντολών που αποτελεί τον κορμό του βρόχου
 - Σκοπός είναι η αύξηση (ή η μείωση) της μεταβλητής ελέγχου του βρόχου κατά ένα συγκεκριμένο ποσό: το βήμα αύξησης/ μείωσης
- Παράδειγμα

```
for (counter = 1; counter <= 10; counter++)  
    printf( "%d\n", counter );
```
- Τυπώνει τους ακέραιους από το ένα ως το 10



Η εντολή επανάληψης for (3/5)





Πρόγραμμα C

```
1  /* Counter-controlled repetition with the for statement */
2  #include <stdio.h>
3
4  /* function main begins program execution */
5  int main()
6  {
7      int counter; /* define counter */
8
9      /* initialization, repetition condition, and increment
10         are all included in the for statement header. */
11     for ( counter = 1; counter <= 10; counter++ ) {
12         printf( "%d\n", counter );
13     } /* end for */
14
15     return 0; /* indicate program ended successfully */
16
17 } /* end function main */
```



Η εντολή επανάληψης for (4/5)

- Οι βρόχοι for μπορούν συνήθως να γραφτούν και σαν βρόχοι while

```
αρχικοποίηση;  
while ( συνθήκη-ελέγχου) {  
    εντολή;  
    αύξηση/ μείωση;  
}
```

- Αρχικοποίηση και αύξηση / μείωση

- Μπορούν να αποτελούνται από μια ομάδα εντολών που διαχωρίζονται με κόμμα (,)
- Παράδειγμα

```
for (i = 0, j = 0; j + i <= 10; j++, i++)  
    printf( "%d\n", j + i );
```



Η εντολή επανάληψης for (5/5)

□ Αριθμητικές εκφράσεις

- Αρχικοποίηση, συνθήκη ελέγχου και αύξηση/ μείωση μπορεί να περιέχουν αριθμητικές εκφράσεις

■ Παράδειγμα:

- Αν το x είναι ίσο με 2 και το y είναι ίσο με 10

```
for ( j = x; j <= 4 * x * y; j += y / x )
```

- ισοδυναμεί με

```
for ( j = 2; j <= 80; j += 5 )
```



Πρόγραμμα C

```
1  /*Summation with for */
2  #include <stdio.h>
3
4  /* function main begins program execution */
5  int main()
6  {
7      int sum = 0; /* initialize sum */
8      int number; /* number to be added to sum */
9
10     for ( number = 2; number <= 100; number += 2 ) {
11         sum += number; /* add number to sum */
12     } /* end for */
13
14     printf( "Sum is %d\n", sum ); /* output sum */
15
16     return 0; /* indicate program ended successfully */
17
18 } /* end function main */
```

```
Sum is 2550
```



Πρόγραμμα C

```
1  /* Calculating compound interest */
2  #include <stdio.h>
3  #include <math.h>
4
5  /* function main begins program execution */
6  int main()
7  {
8      double amount;          /* amount on deposit */
9      double principal = 1000.0; /* starting principal */
10     double rate = .05;      /* interest rate */
11     int year;               /* year counter */
12
13     /* output table column head */
14     printf( "%4s%21s\n", "Year", "Amount on deposit" );
15
16     /* calculate amount on deposit for each of ten years */
17     for ( year = 1; year <= 10; year++ ) {
18
19         /* calculate new amount for specified year */
20         amount = principal * pow( 1.0 + rate, year );
21
22         /* output one table row */
23         printf( "%4d%21.2f\n", year, amount );
24     } /* end for */
25
26     return 0; /* indicate program ended successfully */
27
28 } /* end function main */
```



Πρόγραμμα C

```
Year      Amount on deposit
  1          1050.00
  2          1102.50
  3          1157.63
  4          1215.51
  5          1276.28
  6          1340.10
  7          1407.10
  8          1477.46
  9          1551.33
 10          1628.89
```



Σημερινή διάλεξη

- Σύνοψη 2ης διάλεξης
- Δομές ελέγχου
 - Η εντολή If...Else
 - Η εντολή επανάληψης While
- Διατύπωση αλγορίθμων
 - Επανάληψη ελεγχόμενη από μετρητή
 - Επανάληψη ελεγχόμενη από τιμή φρουρό
 - Ένθετες δομές ελέγχου
 - Σύγκριση “επανάληψης” με μετρητή / τιμή «σημαία»
- Η εντολή επανάληψης for
- Η εντολή πολλαπλής επιλογής switch
- Η εντολή break

Η εντολή πολλαπλής επιλογής switch (1/2)



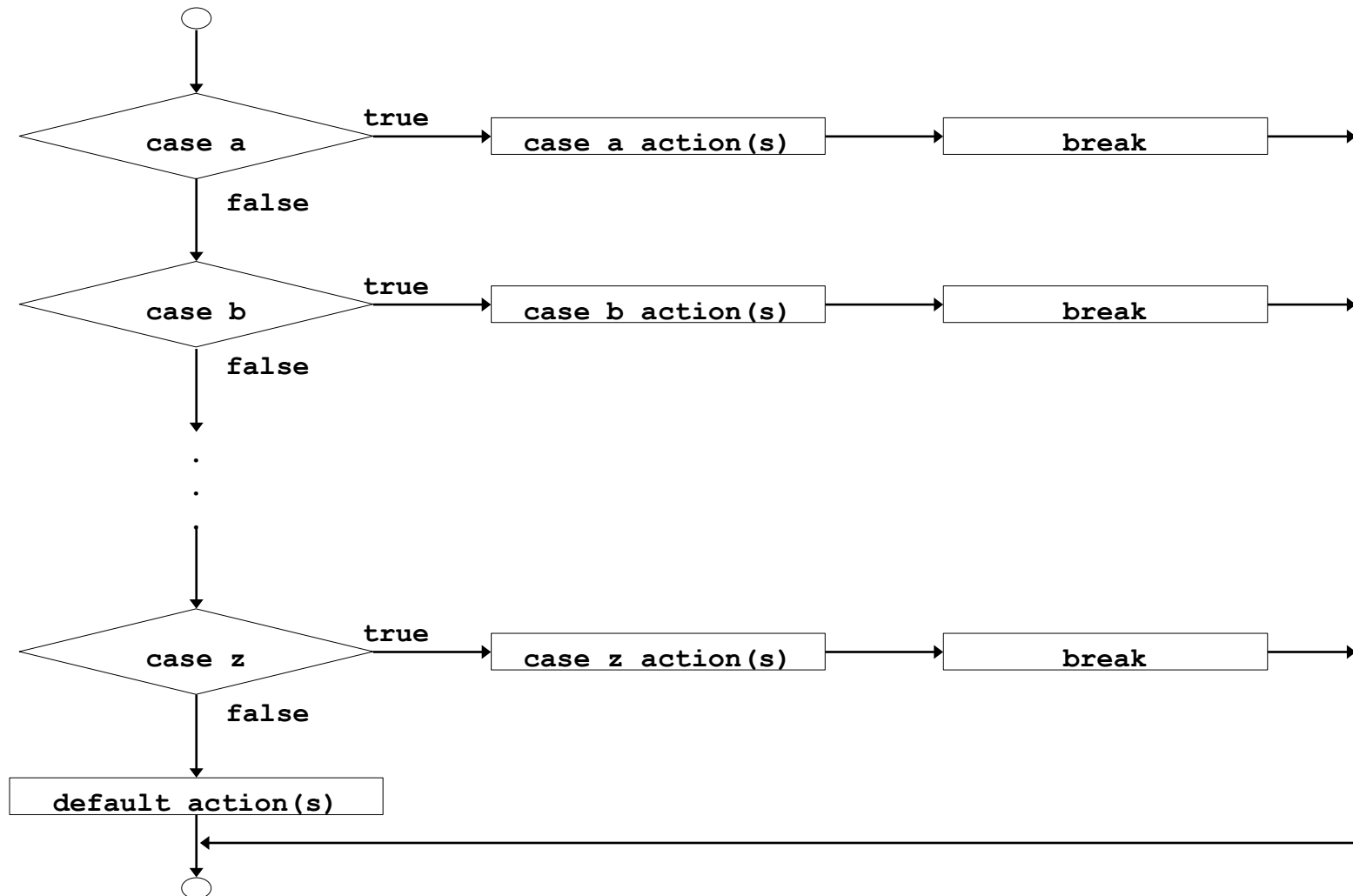
□ switch

- Χρήσιμη όταν μια μεταβλητή ή έκφραση ελέγχεται για όλες τις δυνατές τιμές που μπορεί να πάρει και γίνονται διαφορετικές ενέργειες
- Η τιμή ελέγχεται διαδοχικά έναντι μιας λίστας σταθερών (ακέραιων ή χαρακτήρων)
 - Όταν βρεθεί ένα «ταίριασμα» εκτελείται η αλληλουχία εντολών που σχετίζεται με αυτή την περίπτωση - case μέχρι να βρεθεί μια break
 - default
 - Η αλληλουχία εντολών της εκτελείται αν δεν βρεθεί κανένα «ταίριασμα»
 - Προαιρετική- αν δεν υπάρχει δεν εκτελείται καμία εντολή

Γενική μορφή

```
switch ( τιμή ) {  
    case σταθερά1:  
        εντολές  
        break;  
    case σταθερά2:  
        εντολές  
        break;  
    .  
    .  
    .  
    default:  
        εντολές  
        break;  
}
```

Η εντολή πολλαπλής επιλογής switch (2/2)





Παράδειγμα (1/4)

```
1  /* Counting letter grades */
2  #include <stdio.h>
3
4  /* function main begins program execution */
5  int main()
6  {
7      int grade;      /* one grade */
8      int aCount = 0; /* number of As */
9      int bCount = 0; /* number of Bs */
10     int cCount = 0; /* number of Cs */
11     int dCount = 0; /* number of Ds */
12     int fCount = 0; /* number of Fs */
13
14     printf( "Enter the letter grades.\n" );
15     printf( "Enter the EOF character to end input.\n" );
16
17     /* loop until user types end-of-file key sequence */
18     while ( ( grade = getchar() ) != EOF ) {
19
20         /* determine which grade was input */
21         switch ( grade ) { /* switch nested in while */
22
23             case 'A':      /* grade was uppercase A */
24             case 'a':      /* or lowercase a */
25                 ++aCount; /* increment aCount */
26                 break;    /* necessary to exit switch */
27
```



Παράδειγμα (2/4)

```
28     case 'B':    /* grade was uppercase B */
29     case 'b':    /* or lowercase b */
30         ++bCount; /* increment bCount */
31         break;   /* exit switch */
32
33     case 'C':    /* grade was uppercase C */
34     case 'c':    /* or lowercase c */
35         ++cCount; /* increment cCount */
36         break;   /* exit switch */
37
38     case 'D':    /* grade was uppercase D */
39     case 'd':    /* or lowercase d */
40         ++dCount; /* increment dCount */
41         break;   /* exit switch */
42
43     case 'F':    /* grade was uppercase F */
44     case 'f':    /* or lowercase f */
45         ++fCount; /* increment fCount */
46         break;   /* exit switch */
47
48     case '\n':   /* ignore newlines, */
49     case '\t':   /* tabs, */
50     case ' ':    /* and spaces in input */
51         break;   /* exit switch */
52
```



Παράδειγμα (3/4)

```
53     default:    /* catch all other characters */
54         printf( "Incorrect letter grade entered." );
55         printf( " Enter a new grade.\n" );
56         break;  /* optional; will exit switch anyway */
57     } /* end switch */
58
59 } /* end while */
60
61 /* output summary of results */
62 printf( "\nTotals for each letter grade are:\n" );
63 printf( "A: %d\n", aCount ); /* display number of A grades */
64 printf( "B: %d\n", bCount ); /* display number of B grades */
65 printf( "C: %d\n", cCount ); /* display number of C grades */
66 printf( "D: %d\n", dCount ); /* display number of D grades */
67 printf( "F: %d\n", fCount ); /* display number of F grades */
68
69 return 0; /* indicate program ended successfully */
70
71 } /* end function main */
```



Παράδειγμα (4/4)

```
Enter the letter grades.  
Enter the EOF character to end input.  
a  
b  
c  
C  
A  
d  
f  
C  
E  
Incorrect letter grade entered. Enter a new grade.  
D  
A  
b  
^Z  
  
Totals for each letter grade are:  
A: 3  
B: 2  
C: 3  
D: 2  
F: 1
```



Σημερινή διάλεξη

- Σύνοψη 2ης διάλεξης
- Δομές ελέγχου
 - Η εντολή If...Else
 - Η εντολή επανάληψης While
- Διατύπωση αλγορίθμων
 - Επανάληψη ελεγχόμενη από μετρητή
 - Επανάληψη ελεγχόμενη από τιμή φρουρό
 - Ένθετες δομές ελέγχου
 - Σύγκριση “επανάληψης” με μετρητή / τιμή «σημαία»
- Η εντολή επανάληψης for
- Η εντολή πολλαπλής επιλογής switch
- Η εντολή break



Η εντολή `break`

- Προκαλεί άμεση έξοδο από μια εντολή `while`, `for`, `do...while` ή `switch`
- Η εκτέλεση του προγράμματος συνεχίζει με την αμέσως επόμενη εντολή που ακολουθεί τον κορμό μιας από τις παραπάνω εντολές
- Η εντολή `break` χρησιμοποιείται συνήθως
 - Για την πρόωρη έξοδο από ένα βρόχο
 - Για την παράλειψη του υπόλοιπου τμήματος μιας εντολής `switch`



Παράδειγμα

```
1  /* Using the break statement in a for statement */
2  #include <stdio.h>
3
4  /* function main begins program execution */
5  int main()
6  {
7      int x; /* counter */
8
9      /* loop 10 times */
10     for ( x = 1; x <= 10; x++ ) {
11
12         /* if x is 5, terminate loop */
13         if ( x == 5 ) {
14             break; /* break loop only if x is 5 */
15         } /* end if */
16
17         printf( "%d ", x ); /* display value of x */
18     } /* end for */
19
20     printf( "\nBroke out of loop at x == %d\n", x );
21
22     return 0; /* indicate program ended successfully */
23
24 } /* end function main */
```

```
1 2 3 4
Broke out of loop at x == 5
```



Η εντολή `continue`

- Παραλείπει τις υπόλοιπες εντολές του κορμού μιας εντολής `while`, `for` ή `do...while`
 - Συνεχίζει με την επόμενη επανάληψη του βρόχου
- `while` και `do...while`
 - Ο έλεγχος της συνθήκης επανάληψης γίνεται αμέσως μετά την εκτέλεση της εντολής `continue`
- `for`
 - Εκτελείται η αύξηση / μείωση και στη συνέχεια γίνεται ο έλεγχος της συνθήκης επανάληψης



Παράδειγμα

```
1  /* Using the continue statement in a for statement */
2  #include <stdio.h>
3
4  /* function main begins program execution */
5  int main()
6  {
7      int x; /* counter */
8
9      /* loop 10 times */
10     for ( x = 1; x <= 10; x++ ) {
11
12         /* if x is 5, continue with next iteration of loop */
13         if ( x == 5 ) {
14             continue; /* skip remaining code in loop body */
15         } /* end if */
16
17         printf( "%d ", x ); /* display value of x */
18     } /* end for */
19
20     printf( "\nUsed continue to skip printing the value 5\n" );
21
22     return 0; /* indicate program ended successfully */
23
24 } /* end function main */
```

```
1 2 3 4 6 7 8 9 10
Used continue to skip printing the value 5
```



Ερωτήσεις...

