

Διάλεξη 4η: Εντολές Ελέγχου Ροής

Τμήμα Επιστήμης Υπολογιστών, Πανεπιστήμιο Κρήτης

Εισαγωγή στην Επιστήμη Υπολογιστών

Βασίζεται σε διαφάνειες του Κ. Παναγιωτάκη



Είδη εντολών

- Θυμηθείτε τη μηχανή von Neumann
 - Φορτώνει μια εντολή
 - την εκτελεί
 - αποφασίζει ποιά είναι η επόμενη
 - Κ.Ο.Κ.
- Αυτό λέγεται έλεγχος ροής (flow of control ή control-flow) του προγράμματος
- Όλα τα προγράμματα C χρησιμοποιούν 3 είδη εντολών για να ελέγξουν τη ροή του προγράμματος
 - Σειριακές (sequential) εντολές, εκτελείται η μία μετά την άλλη
 - `x = 42; z = f(x); ...`
 - Εντολές απόφασης (conditional), αποφασίζουν τι θα εκτελεστεί, υπό συνθήκη
 - `if (x == 0) { y = 1; } else { y = 2; }`
 - `switch (x) { case 1: ... case 2: ... default: ... }`
 - Εντολές επανάληψης, εκτελούν πολλές φορές την ίδια λειτουργία
 - `for (i = 0; i < 10; i++) { ... }`
 - `while (f(x) < 2) { ... }`



Εντολές υπό συνθήκη: `if`

- Εντολή `if`: ελέγχει μια συνθήκη και αποφασίζει ποιές εντολές θα εκτελεστούν

Γενική μορφή

```
if (expression) {  
    statements  
} else {  
    statements  
}
```



Εντολές υπό συνθήκη: `if`

- Εντολή `if`: ελέγχει μια συνθήκη και αποφασίζει ποιες εντολές θα εκτελεστούν

Γενική μορφή

```
if (expression) {  
    statements  
}  
else {  
    statements  
}
```

Αρχικά υπολογίζεται η τιμή της έκφρασης (expression) ως ακέραιος αριθμός, και συγκρίνεται με το 0



Εντολές υπό συνθήκη: `if`

- Εντολή `if`: ελέγχει μια συνθήκη και αποφασίζει ποιές εντολές θα εκτελεστούν

Γενική μορφή

```
if (expression) {  
    statements  
}  
else {  
    statements  
}
```

Εντολές (statements) που εκτελούνται μόνο αν η έκφραση έχει τιμή διαφορετική από 0



Εντολές υπό συνθήκη: `if`

- Εντολή `if`: ελέγχει μια συνθήκη και αποφασίζει ποιές εντολές θα εκτελεστούν

Γενική μορφή

```
if (expression) {  
    statements  
}  
else {  
    statements  
}
```

Εντολές που εκτελούνται
μόνο αν η έκφραση έχει τιμή
ίση με 0



Εντολές υπό συνθήκη: `if`

- Μπορούμε να παραλείψουμε την περίπτωση `else`

Παράδειγμα 1

```
if (x == 0) {  
    x++;  
}  
x++;
```

- Μπορούμε να “φωλιάσουμε” τα `if`

Παράδειγμα 2

```
if (x == 0) {  
    x++;  
} else if (x == 1) {  
    x = x + 2;  
} else {  
    x = x - 1;  
}
```



Εντολές υπό συνθήκη: `if`

- Μπορούμε να παραλείψουμε την περίπτωση `else`

Παράδειγμα 1 Εκτελείται μόνο αν το `x` είναι 0

```
if (x == 0) {  
    x++;  
}  
x++;
```

- Μπορούμε να “φωλιάσουμε” τα `if`

Παράδειγμα 2

```
if (x == 0) {  
    x++;  
} else if (x == 1) {  
    x = x + 2;  
} else {  
    x = x - 1;  
}
```



Εντολές υπό συνθήκη: `if`

- Μπορούμε να παραλείψουμε την περίπτωση `else`

Παράδειγμα 1

```
if (x == 0) {  
    x++;  
}  
x++;
```

Εκτελείται πάντα, μετά το `if`

- Μπορούμε να “φωλιάσουμε” τα `if`

Παράδειγμα 2

```
if (x == 0) {  
    x++;  
} else if (x == 1) {  
    x = x + 2;  
} else {  
    x = x - 1;  
}
```



Εντολές υπό συνθήκη: `if`

- Μπορούμε να παραλείψουμε την περίπτωση `else`

Παράδειγμα 1

```
if (x == 0) {  
    x++;  
}  
x++;
```

- Μπορούμε να “φωλιάσουμε” τα `if`

Παράδειγμα 2

```
if (x == 0) {  
    x++;  
} else if (x == 1) {  
    x = x + 2;  
} else {  
    x = x - 1;  
}
```



Εντολές υπό συνθήκη: `if`

- Μπορούμε να παραλείψουμε την περίπτωση `else`

Παράδειγμα 1

```
if (x == 0) {  
    x++;  
}  
x++;
```

- Μπορούμε να “φωλιάσουμε” τα `if`

Παράδειγμα 2 Εκτελείται μόνο αν το `x` είναι 0

```
if (x == 0) {  
    x++;  
} else if (x == 1) {  
    x = x + 2;  
} else {  
    x = x - 1;  
}
```



Εντολές υπό συνθήκη: `if`

- Μπορούμε να παραλείψουμε την περίπτωση `else`

Παράδειγμα 1

```
if (x == 0) {  
    x++;  
}  
x++;
```

- Μπορούμε να “φωλιάσουμε” τα `if`

Παράδειγμα 2

```
if (x == 0)  
    x++;  
} else if (x == 1)  
    x = x + 2;  
} else {  
    x = x - 1;  
}
```

Εκτελείται μόνο αν το `x` είναι
1



Εντολές υπό συνθήκη: `if`

- Μπορούμε να παραλείψουμε την περίπτωση `else`

Παράδειγμα 1

```
if (x == 0) {  
    x++;  
}  
x++;
```

- Μπορούμε να “φωλιάσουμε” τα `if`

Παράδειγμα 2

```
if (x == 0) {  
    x++;  
} else if (x == 1) {  
    x = x + 2;  
} else {  
    x = x - 1;  
}
```

Εκτελείται σε κάθε άλλη περίπτωση



Παράδειγμα: `if`

`if.c`

```
#include <stdio.h>

int main() {
    char p;
    float x, y, res;
    scanf("%f %c %f", &x, &p, &y);
    if (p == '+') {
        res = x + y;
    } else if (p == '-') {
        res = x - y;
    } else if (p == '*') {
        res = x * y;
    } else if (p == '/') {
        res = x / y;
    } else {
        printf("Bad input\n");
        return -1;
    }
    printf("%f %c %f = %f\n", x, p, y, res);
    return 0;
}
```



Παράδειγμα: `if`

`if.c`

```
#include <stdio.h>

int main() {
    char p;
    float x, y, res;
    scanf("%f %c %f", &x, &p, &y);
    if (p == '+') {
        res = x + y;
    } else if (p == '-') {
        res = x - y;
    } else if (p == '*') {
        res = x * y;
    } else if (p == '/') {
        res = x / y;
    } else {
        printf("Bad input\n");
        return -1;
    }
    printf("%f %c %f = %f\n", x, p, y, res);
    return 0;
}
```

Διαβάζει έναν αριθμό στο `x`,
ένα σύμβολο (πράξη) `p`, και
ένα δεύτερο αριθμό `y`



Παράδειγμα: `if`

if.c

```
#include <stdio.h>
```

```
int main() {
```

```
    char p;
```

```
    float x, y, res;
```

```
    scanf("%f %c %f", &x, &p, &y);
```

```
    if (p == '+') {
```

```
        res = x + y;
```

```
    } else if (p == '-') {
```

```
        res = x - y;
```

```
    } else if (p == '*') {
```

```
        res = x * y;
```

```
    } else if (p == '/') {
```

```
        res = x / y;
```

```
    } else {
```

```
        printf("Bad input\n");
```

```
        return -1;
```

```
    }
```

```
    printf("%f %c %f = %f\n", x, p, y, res);
```

```
    return 0;
```

```
}
```

Αν το σύμβολο είναι το
σημείο της πρόσθεσης ...



Παράδειγμα: `if`

if.c

```
#include <stdio.h>
```

```
int main() {
```

```
    char p;
```

```
    float x, y, res;
```

```
    scanf("%f %c %f", &x, &p, &y);
```

```
    if (p == '+') {
```

```
        res = x + y;
```

```
    } else if (p == '-') {
```

```
        res = x - y;
```

```
    } else if (p == '*') {
```

```
        res = x * y;
```

```
    } else if (p == '/') {
```

```
        res = x / y;
```

```
    } else {
```

```
        printf("Bad input\n");
```

```
        return -1;
```

```
    }
```

```
    printf("%f %c %f = %f\n", x, p, y, res);
```

```
    return 0;
```

```
}
```

...τότε κάνει την πρόσθεση
και αποθηκεύουμε το
αποτέλεσμα στη μεταβλητή
`res`



Παράδειγμα: `if`

if.c

```
#include <stdio.h>
```

```
int main() {
```

```
    char p;
```

```
    float x, y, res;
```

```
    scanf("%f %c %f", &x, &p, &y); αλλιώς, αν είναι το σύμβολο  
    της αφαίρεσης ...
```

```
    if (p == '+') {
```

```
        res = x + y;
```

```
    } else if (p == '-') {
```

```
        res = x - y;
```

```
    } else if (p == '*') {
```

```
        res = x * y;
```

```
    } else if (p == '/') {
```

```
        res = x / y;
```

```
    } else {
```

```
        printf("Bad input\n");
```

```
        return -1;
```

```
    }
```

```
    printf("%f %c %f = %f\n", x, p, y, res);
```

```
    return 0;
```

```
}
```



Παράδειγμα: `if`

if.c

```
#include <stdio.h>
```

```
int main() {
```

```
    char p;
```

```
    float x, y, res;
```

```
    scanf("%f %c %f", &x, &p, &y);
```

```
    if (p == '+') {
```

```
        res = x + y;
```

```
    } else if (p == '-') {
```

```
        res = x - y;
```

```
    } else if (p == '*') {
```

```
        res = x * y;
```

```
    } else if (p == '/') {
```

```
        res = x / y;
```

```
    } else {
```

```
        printf("Bad input\n");
```

```
        return -1;
```

```
    }
```

```
    printf("%f %c %f = %f\n", x, p, y, res);
```

```
    return 0;
```

```
}
```

...τότε κάνει αφαίρεση



Παράδειγμα: `if`

if.c

```
#include <stdio.h>
```

```
int main() {
```

```
    char p;
```

```
    float x, y, res;
```

```
    scanf("%f %c %f", &x, &p, &y);
```

```
    if (p == '+') {
```

```
        res = x + y;
```

```
    } else if (p == '-') {
```

```
        res = x - y;
```

```
    } else if (p == '*') {
```

```
        res = x * y;
```

```
    } else if (p == '/') {
```

```
        res = x / y;
```

```
    } else {
```

```
        printf("Bad input\n");
```

```
        return -1;
```

```
    }
```

```
    printf("%f %c %f = %f\n", x, p, y, res);
```

```
    return 0;
```

```
}
```

αλλιώς, αν είναι το σύμβολο
του πολλαπλασιασμού ...



Παράδειγμα: `if`

if.c

```
#include <stdio.h>
```

```
int main() {
```

```
    char p;
```

```
    float x, y, res;
```

```
    scanf("%f %c %f", &x, &p, &y);
```

```
    if (p == '+') {
```

```
        res = x + y;
```

```
    } else if (p == '-') {
```

```
        res = x - y;
```

```
    } else if (p == '*') {
```

```
        res = x * y;
```

```
    } else if (p == '/') {
```

```
        res = x / y;
```

```
    } else {
```

```
        printf("Bad input\n");
```

```
        return -1;
```

```
    }
```

```
    printf("%f %c %f = %f\n", x, p, y, res);
```

```
    return 0;
```

```
}
```

...τότε κάνει πολλαπλασιασμό



Παράδειγμα: `if`

if.c

```
#include <stdio.h>

int main() {
    char p;
    float x, y, res;
    scanf("%f %c %f", &x, &p, &y);
    if (p == '+') {
        res = x + y;
    } else if (p == '-') {
        res = x - y;
    } else if (p == '*') {
        res = x * y;
    } else if (p == '/') {
        res = x / y;
    } else {
        printf("Bad input\n");
        return -1;
    }
    printf("%f %c %f = %f\n", x, p, y, res);
    return 0;
}
```

αλλιώς, αν είναι το σύμβολο
της διαίρεσης ...



Παράδειγμα: `if`

if.c

```
#include <stdio.h>
```

```
int main() {
```

```
    char p;
```

```
    float x, y, res;
```

```
    scanf("%f %c %f", &x, &p, &y);
```

```
    if (p == '+') {
```

```
        res = x + y;
```

```
    } else if (p == '-') {
```

```
        res = x - y;
```

```
    } else if (p == '*') {
```

```
        res = x * y;
```

```
    } else if (p == '/') {
```

```
        res = x / y;
```

```
    } else {
```

```
        printf("Bad input\n");
```

```
        return -1;
```

```
    }
```

```
    printf("%f %c %f = %f\n", x, p, y, res);
```

```
    return 0;
```

```
}
```

...τότε κάνει διαίρεση



Παράδειγμα: `if`

if.c

```
#include <stdio.h>
```

```
int main() {
```

```
    char p;
```

```
    float x, y, res;
```

```
    scanf("%f %c %f", &x, &p, &y);
```

```
    if (p == '+') {
```

```
        res = x + y;
```

```
    } else if (p == '-') {
```

```
        res = x - y;
```

```
    } else if (p == '*') {
```

```
        res = x * y;
```

```
    } else if (p == '/') {
```

```
        res = x / y;
```

```
    } else {
```

```
        printf("Bad input\n");
```

```
        return -1;
```

```
    }
```

```
    printf("%f %c %f = %f\n", x, p, y, res);
```

```
    return 0;
```

```
}
```

αλλιώς αν δεν είναι κανένα
από τα παραπάνω ...



Παράδειγμα: `if`

if.c

```
#include <stdio.h>

int main() {
    char p;
    float x, y, res;
    scanf("%f %c %f", &x, &p, &y);
    if (p == '+') {
        res = x + y;
    } else if (p == '-') {
        res = x - y;
    } else if (p == '*') {
        res = x * y;
    } else if (p == '/') {
        res = x / y;
    } else {
        printf("Bad input\n");
        return -1;
    }
    printf("%f %c %f = %f\n", x, p, y, res);
    return 0;
}
```

...τότε τυπώνει μήνυμα
λάθους



Παράδειγμα: `if`

if.c

```
#include <stdio.h>

int main() {
    char p;
    float x, y, res;
    scanf("%f %c %f", &x, &p, &y);
    if (p == '+') {
        res = x + y;
    } else if (p == '-') {
        res = x - y;
    } else if (p == '*') {
        res = x * y;
    } else if (p == '/') {
        res = x / y;
    } else {
        printf("Bad input\n");
        return -1;
    }
    printf("%f %c %f = %f\n", x, p, y, res);
    return 0;
}
```

...και επιστρέφει -1



Παράδειγμα: `if`

if.c

```
#include <stdio.h>

int main() {
    char p;
    float x, y, res;
    scanf("%f %c %f", &x, &p, &y);
    if (p == '+') {
        res = x + y;
    } else if (p == '-') {
        res = x - y;
    } else if (p == '*') {
        res = x * y;
    } else if (p == '/') {
        res = x / y;
    } else {
        printf("Bad input\n");
        return -1;
    }
    printf("%f %c %f = %f\n", x, p, y, res);
    return 0;
}
```

Τυπώνει τον πρώτο αριθμό, το σύμβολο της πράξης, τον δεύτερο αριθμό, και το αποτέλεσμα της πράξης

Παράδειγμα: `if`

if.c

```
#include <stdio.h>

int main() {
    char p;
    float x, y, res;
    scanf("%f %c %f", &x, &p, &y);
    if (p == '+') {
        res = x + y;
    } else if (p == '-') {
        res = x - y;
    } else if (p == '*') {
        res = x * y;
    } else if (p == '/') {
        res = x / y;
    } else {
        printf("Bad input\n");
        return -1;
    }
    printf("%f %c %f = %f\n", x, p, y, res);
    return 0;
}
```

Προσοχή: αυτή η γραμμή
εκτελείται μόνο αν το
πρόγραμμα φτάσει εδώ
(δεν έχει επιστρέψει -1)

Εντολές υπό συνθήκη: switch

- Εντολή **switch**: επιλέγει τις εντολές που θα εκτελεστούν ανάλογα με μια τιμή

Γενική μορφή

```
switch (expression) {  
  case constant1:  
    statements;  
  break;  
  case constant2:  
    statements;  
  break;  
  default:  
    statements;  
}
```



Εντολές υπό συνθήκη: switch

- Εντολή **switch**: επιλέγει τις εντολές που θα εκτελεστούν ανάλογα με μια τιμή

Γενική μορφή

```
switch (expression) {  
  case constant1:  
    statements;  
  break;  
  case constant2:  
    statements;  
  break;  
  default:  
    statements;  
}
```

Αρχικά υπολογίζεται η τιμή της έκφρασης (expression)



Εντολές υπό συνθήκη: switch

- Εντολή **switch**: επιλέγει τις εντολές που θα εκτελεστούν ανάλογα με μια τιμή

Γενική μορφή

```
switch (expression) {  
  case constant1:  
    statements;  
  break;  
  case constant2:  
    statements;  
  break;  
  default:  
    statements;  
}
```

Συγκρίνεται η τιμή που υπολογίστηκε με τη σταθερά **constant1**



Εντολές υπό συνθήκη: switch

- Εντολή **switch**: επιλέγει τις εντολές που θα εκτελεστούν ανάλογα με μια τιμή

Γενική μορφή

```
switch (expression) {  
  case constant1:  
    statements;  
    break;  
  case constant2:  
    statements;  
    break;  
  default:  
    statements;  
}
```

Αν είναι ίση, εκτελούνται οι
εντολές που ακολουθούν το
case



Εντολές υπό συνθήκη: switch

- Εντολή **switch**: επιλέγει τις εντολές που θα εκτελεστούν ανάλογα με μια τιμή

Γενική μορφή

```
switch (expression) {  
  case constant1:  
    statements;  
    break;  
  case constant2:  
    statements;  
    break;  
  default:  
    statements;  
}
```

Αν υπάρχει **break** στο τέλος των εντολών η εκτέλεση συνεχίζει μετά το **switch**



Εντολές υπό συνθήκη: switch

- Εντολή **switch**: επιλέγει τις εντολές που θα εκτελεστούν ανάλογα με μια τιμή

Γενική μορφή

```
switch (expression) {  
  case constant1:  
    statements;  
    break;  
  case constant2:  
    statements;  
    break;  
  default:  
    statements;  
}
```

Αν η τιμή της έκφρασης του **switch** δεν είναι ίση με τη σταθερά **constant1**, τότε συγκρίνεται με τη σταθερά **constant2**



Εντολές υπό συνθήκη: switch

- Εντολή **switch**: επιλέγει τις εντολές που θα εκτελεστούν ανάλογα με μια τιμή

Γενική μορφή

```
switch (expression) {  
  case constant1:  
    statements;  
    break;  
  case constant2:  
    statements;  
    break;  
  default:  
    statements;  
}
```

Αν είναι ίση, εκτελούνται οι εντολές που ακολουθούν...



Εντολές υπό συνθήκη: switch

- Εντολή **switch**: επιλέγει τις εντολές που θα εκτελεστούν ανάλογα με μια τιμή

Γενική μορφή

```
switch (expression) {  
  case constant1:  
    statements;  
    break;  
  case constant2:  
    statements;  
    break;  
  default:  
    statements;  
}
```

Αν η τιμή που υπολογίστηκε
δεν είναι ίση με καμία
σταθερά, εκτελούνται οι
εντολές που ακολουθούν το
default



Παράδειγμα: switch

switch.c

```
#include <stdio.h>
int main(void) {
    char p;
    float x, y, res;
    scanf("%f %c %f", &x, &p, &y);
    switch(p) {
        case '+':
            res = x + y; break;
        case '-':
            res = x - y; break;
        case '*':
            res = x * y; break;
        case '/':
            res = x / y; break;
        default:
            printf("Bad input\n");
            return -1;
    }
    printf("%f %c %f = %f\n", x, p, y, res);
    return 0;
}
```



Παράδειγμα: switch

switch.c

```
#include <stdio.h>
int main(void) {
    char p;
    float x, y, res;
    scanf("%f %c %f", &x, &p, &y);
    switch(p) {
        case '+':
            res = x + y; break;
        case '-':
            res = x - y; break;
        case '*':
            res = x * y; break;
        case '/':
            res = x / y; break;
        default:
            printf("Bad input\n");
            return -1;
    }
    printf("%f %c %f = %f\n", x, p, y, res);
    return 0;
}
```

Διάβασε από το πληκτρολόγιο
έναν αριθμό κινητής
υποδιαστολής και αποθήκευσέ
τον στη μεταβλητή **x**, ένα
χαρακτήρα στην **p** και ένα
δεύτερο αριθμό στην **y**



Παράδειγμα: switch

switch.c

```
#include <stdio.h>
int main(void) {
    char p;
    float x, y, res;
    scanf("%f %c %f", &x, &p, &y);
    switch(p) {
        case '+':
            res = x + y; break;
        case '-':
            res = x - y; break;
        case '*':
            res = x * y; break;
        case '/':
            res = x / y; break;
        default:
            printf("Bad input\n");
            return -1;
    }
    printf("%f %c %f = %f\n", x, p, y, res);
    return 0;
}
```

Έλεγξε τον χαρακτήρα που περιέχει η μεταβλητή **p**



Παράδειγμα: switch

switch.c

```
#include <stdio.h>
int main(void) {
    char p;
    float x, y, res;
    scanf("%f %c %f", &x, &p, &y);
    switch(p) {
        case '+':
            res = x + y; break;
        case '-':
            res = x - y; break;
        case '*':
            res = x * y; break;
        case '/':
            res = x / y; break;
        default:
            printf("Bad input\n");
            return -1;
    }
    printf("%f %c %f = %f\n", x, p, y, res);
    return 0;
}
```

Αν είναι ο χαρακτήρας + ...



Παράδειγμα: switch

switch.c

```
#include <stdio.h>
int main(void) {
    char p;
    float x, y, res;
    scanf("%f %c %f", &x, &p, &y);
    switch(p) {
        case '+':
            res = x + y; break;
        case '-':
            res = x - y; break;
        case '*':
            res = x * y; break;
        case '/':
            res = x / y; break;
        default:
            printf("Bad input\n");
            return -1;
    }
    printf("%f %c %f = %f\n", x, p, y, res);
    return 0;
}
```

... τότε πρόσθεσε τους δύο αριθμούς x και y και αποθήκευσε το αποτέλεσμα στη μεταβλητή res ...



Παράδειγμα: switch

switch.c

```
#include <stdio.h>
int main(void) {
    char p;
    float x, y, res;
    scanf("%f %c %f", &x, &p, &y);
    switch(p) {
        case '+':
            res = x + y; break;
        case '-':
            res = x - y; break;
        case '*':
            res = x * y; break;
        case '/':
            res = x / y; break;
        default:
            printf("Bad input\n");
            return -1;
    }
    printf("%f %c %f = %f\n", x, p, y, res);
    return 0;
}
```

... και βγες από τη switch



Παράδειγμα: switch

switch.c

```
#include <stdio.h>
int main(void) {
    char p;
    float x, y, res;
    scanf("%f %c %f", &x, &p, &y);
    switch(p) {
        case '+':
            res = x + y; break;
        case '-':
            res = x - y; break;
        case '*':
            res = x * y; break;
        case '/':
            res = x / y; break;
        default:
            printf("Bad input\n");
            return -1;
    }
    printf("%f %c %f = %f\n", x, p, y, res);
    return 0;
}
```

Αν είναι ο χαρακτήρας - ...



Παράδειγμα: switch

switch.c

```
#include <stdio.h>
int main(void) {
    char p;
    float x, y, res;
    scanf("%f %c %f", &x, &p, &y);
    switch(p) {
        case '+':
            res = x + y; break;
        case '-':
            res = x - y; break;
        case '*':
            res = x * y; break;
        case '/':
            res = x / y; break;
        default:
            printf("Bad input\n");
            return -1;
    }
    printf("%f %c %f = %f\n", x, p, y, res);
    return 0;
}
```

... τότε αφάιρεσε τους δύο αριθμούς x και y ...



Παράδειγμα: switch

switch.c

```
#include <stdio.h>
int main(void) {
    char p;
    float x, y, res;
    scanf("%f %c %f", &x, &p, &y);
    switch(p) {
        case '+':
            res = x + y; break;
        case '-':
            res = x - y; break;
        case '*':
            res = x * y; break;
        case '/':
            res = x / y; break;
        default:
            printf("Bad input\n");
            return -1;
    }
    printf("%f %c %f = %f\n", x, p, y, res);
    return 0;
}
```

... και βγες από τη switch



Παράδειγμα: switch

switch.c

```
#include <stdio.h>
int main(void) {
    char p;
    float x, y, res;
    scanf("%f %c %f", &x, &p, &y);
    switch(p) {
        case '+':
            res = x + y; break;
        case '-':
            res = x - y; break;
        case '*':
            res = x * y; break;
        case '/':
            res = x / y; break;
        default:
            printf("Bad input\n");
            return -1;
    }
    printf("%f %c %f = %f\n", x, p, y, res);
    return 0;
}
```

Αν είναι ο χαρακτήρας * τότε
πολλαπλασιάσέ τους



Παράδειγμα: switch

switch.c

```
#include <stdio.h>
int main(void) {
    char p;
    float x, y, res;
    scanf("%f %c %f", &x, &p, &y);
    switch(p) {
        case '+':
            res = x + y; break;
        case '-':
            res = x - y; break;
        case '*':
            res = x * y; break;
        case '/':
            res = x / y; break;
        default:
            printf("Bad input\n");
            return -1;
    }
    printf("%f %c %f = %f\n", x, p, y, res);
    return 0;
}
```

Αν είναι ο χαρακτήρας / τότε
διαίρεσέ τους



Παράδειγμα: switch

switch.c

```
#include <stdio.h>
int main(void) {
    char p;
    float x, y, res;
    scanf("%f %c %f", &x, &p, &y);
    switch(p) {
        case '+':
            res = x + y; break;
        case '-':
            res = x - y; break;
        case '*':
            res = x * y; break;
        case '/':
            res = x / y; break;
        default:
            printf("Bad input\n");
            return -1;
    }
    printf("%f %c %f = %f\n", x, p, y, res);
    return 0;
}
```

Αν δεν είναι τίποτε από τα παραπάνω, τύπωσε ένα μήνυμα λάθους και επέστρεψε -1



Παράδειγμα: switch

switch.c

```
#include <stdio.h>
int main(void) {
    char p;
    float x, y, res;
    scanf("%f %c %f", &x, &p, &y);
    switch(p) {
        case '+':
            res = x + y; break;
        case '-':
            res = x - y; break;
        case '*':
            res = x * y; break;
        case '/':
            res = x / y; break;
        default:
            printf("Bad input\n");
            return -1;
    }
    printf("%f %c %f = %f\n", x, p, y, res);
    return 0;
}
```

Μετά τη `switch`,
τύπωσε την πράξη
και το αποτέλεσμα
της, και επέστρεψε
0



Εκφράσεις υπό συνθήκη: ? :

- Έκφραση “?:”: επιλέγει ανάμεσα σε δύο εκφράσεις, ανάλογα με μια συνθήκη

Γενική μορφή

condition ? expression1 : expression2

- Παραδείγματα
 - Η έκφραση `1 ? a : b` ισοδυναμεί με την έκφραση `a`
 - Η έκφραση `0 ? a : b` ισοδυναμεί με την έκφραση `b`
 - Η εντολή `x = (x == 1) ? 3 : 42;` ισοδυναμεί με την εντολή

```
if (x == 1) {  
    x = 3;  
} else {  
    x = 42;  
}
```



Εκφράσεις υπό συνθήκη: ? :

- Έκφραση “?:”: επιλέγει ανάμεσα σε δύο εκφράσεις, ανάλογα με μια συνθήκη

Γενική μορφή

condition : expression1 : expression2

Αρχικά υπολογίζεται η τιμή της έκφρασης συνθήκης (condition)

- Παραδείγματα
 - Η έκφραση $1 ? a : b$ ισοδυναμεί με την έκφραση a
 - Η έκφραση $0 ? a : b$ ισοδυναμεί με την έκφραση b
 - Η εντολή $x = (x == 1) ? 3 : 42;$ ισοδυναμεί με την εντολή

```
if (x == 1) {  
    x = 3;  
} else {  
    x = 42;  
}
```



Εκφράσεις υπό συνθήκη: ?:

- Έκφραση “?:”: επιλέγει ανάμεσα σε δύο εκφράσεις, ανάλογα με μια συνθήκη. Αν είναι διάφορη του 0, τότε επιστρέφεται το αποτέλεσμα της **expression1**

Γενική μορφή

condition ? expression1 : expression2

expression1

- Παραδείγματα

- Η έκφραση **1 ? a : b** ισοδυναμεί με την έκφραση **a**
- Η έκφραση **0 ? a : b** ισοδυναμεί με την έκφραση **b**
- Η εντολή **x = (x == 1) ? 3 : 42;** ισοδυναμεί με την εντολή

```
if (x == 1) {  
    x = 3;  
} else {  
    x = 42;  
}
```



Εκφράσεις υπό συνθήκη: ? :

- Έκφραση “?:”: επιλέγει ανάμεσα σε δύο εκφράσεις, ανάλογα με μια συνθήκη

Γενική μορφή

condition ? expression1 : expression2

Αν είναι ίση με 0, τότε επιστρέφεται το αποτέλεσμα της **expression2**

- Παραδείγματα
 - Η έκφραση `1 ? a : b` ισοδυναμεί με την έκφραση `a`
 - Η έκφραση `0 ? a : b` ισοδυναμεί με την έκφραση `b`
 - Η εντολή `x = (x == 1) ? 3 : 42;` ισοδυναμεί με την εντολή

```
if (x == 1) {  
    x = 3;  
} else {  
    x = 42;  
}
```



- `if (x = 42)` αντί για `if (x == 42)`
- Στην εντολή `switch`, δεν υπάρχει `break` στο τέλος του `case`
- Ξεχάσατε να βάλετε `{` και `}`
- Λάθος στις προτεραιότητες των τελεστών, ή ξεχάσατε τις παρενθέσεις
- Για αποφυγή λαθών: καλή μορφοποίηση του κώδικα, σχόλια



Μορφοποίηση κώδικα

- Σχόλια στις δηλώσεις συναρτήσεων και μεταβλητών
 - Τι υπολογίζει/επιστρέφει η συνάρτηση; Τι είναι η κάθε παράμετρος;
- Σωστά ονόματα συναρτήσεων και μεταβλητών
- Κατάλληλη χρήση indentation, παρενθέσεων, {}

Παράδειγμα

```
/* Συνάρτηση get_median: Υπολογίζει το μεσαίο από τρεις ακέραιους αριθμούς.
 * Δέχεται τρεις ακέραιους αριθμούς. Επιστρέφει τον μεσαίο αριθμό.
 */
int get_median(int a, int b, int c) {
    int median; /* Κρατάει το μεσαίο από τους τρεις αριθμούς x, y, z */

    if ((b <= a) && (a <= c)) || ((b >= a) && (a >= c)) {
        median = a;
    } else if ((a <= b) && (b <= c)) || ((a >= b) && (b >= c)) {
        median = b;
    } else {
        median = c;
    }
    return median;
}
```

Preprocessor: #define

- Ο preprocessor επεξεργάζεται τον κώδικα πριν τον compiler
- Δέχεται ειδικές οδηγίες (directives)
- Κάθε οδηγία αλλάζει τον κώδικα με κάποιο τρόπο
- Όταν εκτελεστούν όλες οι οδηγίες, πρέπει να προκύπτει σωστό πρόγραμμα
- `#define name definition`: αντικαθιστά όλα τα `name` στο πρόγραμμα με `definition`



Preprocessor: #define

- Ο preprocessor επεξεργάζεται τον κώδικα πριν τον compiler
- Δέχεται ειδικές οδηγίες (directives)
- Κάθε οδηγία αλλάζει τον κώδικα με κάποιο τρόπο
- Όταν εκτελεστούν όλες οι οδηγίες, πρέπει να προκύπτει σωστό πρόγραμμα
- `#define name definition`: αντικαθιστά όλα τα `name` στο πρόγραμμα με `definition`

```
#define N 1000  
#define Pi 3.14159  
#define truth_name(x) ((x) ? "true" : "false")
```

```
x = N;  
circumference = 2 * Pi * radius;  
printf("In C, 1 = %s\n", truth_name(1));
```

Preprocessor: #define

- Ο preprocessor επεξεργάζεται τον κώδικα πριν τον compiler
- Δέχεται ειδικές οδηγίες (directives)
- Κάθε οδηγία αλλάζει τον κώδικα με κάποιο τρόπο
- Όταν εκτελεστούν όλες οι οδηγίες, πρέπει να προκύπτει σωστό πρόγραμμα
- `#define name definition`: αντικαθιστά όλα τα `name` στο πρόγραμμα με `definition`

```
#define N 1000  
#define Pi 3.14159  
#define truth_name(x) ((x) ? "true" : "false")
```

```
x = 1000;  
circumference = 2 * Pi * radius;  
printf("In C, 1 = %s\n", truth_name(1));
```

Preprocessor: #define

- Ο preprocessor επεξεργάζεται τον κώδικα πριν τον compiler
- Δέχεται ειδικές οδηγίες (directives)
- Κάθε οδηγία αλλάζει τον κώδικα με κάποιο τρόπο
- Όταν εκτελεστούν όλες οι οδηγίες, πρέπει να προκύπτει σωστό πρόγραμμα
- `#define name definition`: αντικαθιστά όλα τα `name` στο πρόγραμμα με `definition`

```
#define N 1000  
#define Pi 3.14159  
#define truth_name(x) ((x) ? "true" : "false")
```

```
x = 1000;  
circumference = 2 * 3.14159 * radius;  
printf("In C, 1 = %s\n", truth_name(1));
```

Preprocessor: #define

- Ο preprocessor επεξεργάζεται τον κώδικα πριν τον compiler
- Δέχεται ειδικές οδηγίες (directives)
- Κάθε οδηγία αλλάζει τον κώδικα με κάποιο τρόπο
- Όταν εκτελεστούν όλες οι οδηγίες, πρέπει να προκύπτει σωστό πρόγραμμα
- `#define name definition`: αντικαθιστά όλα τα `name` στο πρόγραμμα με `definition`

```
#define N 1000
#define Pi 3.14159
#define truth_name(x) ((x) ? "true" : "false")

x = 1000;
circumference = 2 * 3.14159 * radius;
printf("In C, 1 = %s\n", ((1) ? "true" : "false"));
```

Preprocessor: #include

- `#include <file.h>`: εισάγει ολόκληρο το αρχείο `file.h` σε αυτό το σημείο του προγράμματος

Παράδειγμα

```
#include <stdio.h>
```

- `#include "file.h"`: ίδιο με το παραπάνω, αλλά ψάχνει αρχικά τον τρέχων κατάλογο (directory) για το αρχείο

Παράδειγμα

```
#include "stdio.h"
```

- Μπορεί να είναι φωλιασμένο: κάθε αρχείο που γίνεται `#include` μπορεί να κάνει κι άλλα αρχεία `#include`, κ.ο.κ.



Παράδειγμα: 2βάθμια εξίσωση

quadratic.c

```
/* Solve a 2nd order polynomial */  
#include <stdio.h>  
#include <math.h>  
/* function declarations */  
float discriminant(float, float, float);  
int solve(float a, float b, float c);
```

```
/* function definitions */
```

```
int main(void) {  
    float a, b, c;  
  
    printf("Please type 3 numbers\n");  
    scanf("%f %f %f",&a,&b,&c);  
    printf("Quadratic equation: "  
           "%fx^2 + (%f)x + (%f) = 0\n",  
           a, b, c);  
    if (a == 0) {  
        printf("This is a 1st degree equation\n");  
    } else {  
        solve(a, b, c);  
    }  
}
```

```
float discriminant(float a, float b, float c) {  
    return (b * b) - (4 * a * c);  
}
```

```
int solve(float a, float b, float c) {  
    float D, x1, x2;
```

```
    D = discriminant(a, b, c);
```

```
    if(D < 0) {  
        printf("Discriminant is negative. "  
               "There are no real solutions\n");
```

```
        return 0;
```

```
    } else if(D == 0) {  
        x1 = -b / (2 * a);  
        printf("One solution: %f\n", x1);  
        return 1;
```

```
    } else {  
        x1 = (-b + sqrt(D)) / (2 * a);  
        x2 = (-b - sqrt(D)) / (2 * a);  
        printf("Two solutions: %f, %f\n", x1, x2);  
        return 2;
```

```
    }
```