ΤΕΤΑΡΤΗ ΔΙΑΛΕΞΗ

Εντολή επανάληψης **for**

for *(αρχική_εντολή; συνθήκη; τελική_εντολή) {*
...
}

Εκτέλεση

1. Εκτελείται η «αρχική_εντολή».
2. Ελέγχεται η «συνθήκη».
   - Αν είναι ψευδής, η ροή συνεχίζει με την πρώτη εντολή μετά το σύμπλεγμα **for**.
   - Αν είναι αληθής, εκτελείται το block εντολών μεταξύ των αγκίστρων '{}'. Αν δεν υπάρξει αλλαγή της ροής στο block εκτελείται η «τελική_εντολή».
3. Αν εκτελέστηκε το block χωρίς αλλαγή ροής, επαναλαμβάνεται το βήμα 2.

Παραδείγματα εντολής **for** (1/2)

- Εκτύπωση στην οθόνη των αριθμών 0-9.

```
for (int i{0}; i < 10; ++i) {
   std::cout << i << '\n';
}
```

- Εκτύπωση των αριθμών 99, 97, 95,..., 3, 1:

```
for (int i{99}; i > 0; i-=2) {
   std::cout << i << ' ';
}
```

- Πλήθος των ακεραίων που είναι πολλαπλάσιοι του 2 ή του 3 στο διάστημα $[5, 108]$:

```
std::size_t k{0};
for (int i{5}; i < 109; ++i) {
   if (i%2 == 0 || i%3 == 0) {
      ++k;
   }
}
```

Παραδείγματα εντολής **for** (2/2)

- Άθροισμα περιττών αριθμών μεταξύ 1 και 9.

```
int s{0};
for (int i{1}; i < 10; i+=2) {
   s += i;
}
```

- Λάθος άθροισμα των αριθμών 0.1, 0.2, 0.3:

```
double s{0.0};
for (double x{0.1}; x <= 0.3; x += 0.1) {
   s += x;
}
```

- Σωστό άθροισμα των αριθμών 0.1, 0.2, 0.3:

```
double s{0.0};
for (int i{1}; i <= 3; ++i) {
   s += 0.1 * i;
}
```

Εντολή επανάληψης `while`

while *(συνθήκη) {*
...
}

Εκτέλεση

1. Ελέγχεται η «συνθήκη»:
   - Αν είναι ψευδής, η ροή συνεχίζει με την πρώτη εντολή μετά το σύμπλεγμα **`while`**.
   - Αν είναι αληθής εκτελείται το block εντολών μεταξύ των αγκίστρων '{}'.
2. Αν δεν υπάρξει αλλαγή της ροής στο block επαναλαμβάνεται η διαδικασία από το βήμα 1. Η τιμή της «συνθήκης» μπορεί να μεταβληθεί κατά την εκτέλεση του block εντολών.

Εντολή επανάληψης **while**

Παράδειγμα

Θέλουμε να υπολογίσουμε στο s το άθροισμα των ακέραιων που δίνει ο χρήστης από το πληκτρολόγιο, έως ότου δώσει τον αριθμό 0:

```
int i;
std::cin >> i;

int s{0};
while (i != 0) {
   s += i;
   std::cin >> i;
}
```

Εντολή επανάληψης `do ... while`

```
do {
...
} while (συνθήκη);
```

Εκτέλεση

1. Εκτελείται το block εντολών μεταξύ των αγκίστρων '{}'.
2. Αν δεν υπήρξε αλλαγή ροής ελέγχεται η «συνθήκη»:
   - Αν είναι ψευδής, η ροή συνεχίζει με την πρώτη εντολή μετά την εντολή.
   - Αν είναι αληθής, επαναλαμβάνεται η διαδικασία από το βήμα 1 (εκτέλεση του block).

   Η τιμή της «συνθήκης» μπορεί να μεταβάλλεται σε κάθε επανάληψη.

Εντολή επανάληψης `do ... while`

Παράδειγμα

Θέλουμε να εξασφαλίσουμε ότι ένας ακέραιος που το πρόγραμμά μας θα διαβάζει από το πληκτρολόγιο είναι θετικός. Αν ο χρήστης δώσει αρνητικό ή μηδέν, το πρόγραμμα να επαναλαμβάνει το διάβασμα.

```
int i;
do {
  std::cout << u8"\n Δώσε θετικό ακέραιο: ";
  std::cin >> i;
} while (i <= 0);
```

Αναφορά (1/2)

Μπορούμε να ορίσουμε μια άλλη, ισοδύναμη αλλά ίσως πιο σύντομη, *ονομασία* για ποσότητα (μεταβλητή, σταθερή, συνάρτηση, κλπ.):

Σύνταξη

τύπος όνομαΑ;
τύπος & όνομαΒ{όνομαΑ};
auto *& όνομαΒ = όνομαΑ;*

Δεν δημιουργείται νέα ποσότητα αλλά μόνο αποκτά νέο όνομα η αρχική.

Παράδειγμα

```
int a{3};
auto b = a; // b = 3
auto & c = a;
b=5; // a = 3
c=7; // a = 7
```

Αναφορά (2/2)

Αν η αρχική ποσότητα είναι σταθερή (**const** ή **constexpr**) πρέπει και η αναφορά να είναι **const**:

Παράδειγμα

```
int constexpr a{5};
int & p{a};   // Λάθος
int const & q{a};
/* Σωστό. Δεν μπορεί να αλλάξει το a μέσω του q */
```

Μπορούμε να ορίσουμε (σταθερή) αναφορά σε σταθερή, μεταβλητή ή έκφραση κατάλληλου τύπου:

Παράδειγμα

```
int const & p{4};
int a{3};
int const & q{a};
int const & r{2*a};
```

Εντολή range **for**

Βρόχος που διατρέχει σύνολο στοιχείων (έστω a):

```
for (auto & x : a) {
... // use x (read/write)
}
```

ή

```
for (auto const & x : a) {
... // use x (read)
}
```

Το x γίνεται διαδοχικά από το πρώτο στο τελευταίο στοιχείο, αναφορά (σταθερή ή όχι) στις τιμές των στοιχείων του a και χρησιμοποιείται στο σώμα. Π.χ.

```
for (auto const & x : {4,5,8,-6}) {
   std::cout << x << '\n';
}
```

Εντολή **break**

- Μπορεί να εμφανιστεί μόνο σε σώμα εντολής **switch** ή βρόχου.
- Η εκτέλεσή της προκαλεί διακοπή της εκτέλεσης της εντολής στην οποία βρίσκεται.

Παράδειγμα

Ανάγνωση θετικού ακέραιου.

```
int i;
while (true) {
   std::cout << u8"\n Δώσε θετικό ακέραιο: ";
   std::cin >> i;
   if (i > 0) {
      break;
   }
   std::cout << u8"Έδωσες αρνητικό\n";
}
// ... use i
```

Εντολή `continue`

- Μπορεί να εμφανιστεί μόνο σε σώμα βρόχου.
- Η εκτέλεσή της προκαλεί μετακίνηση της ροής εκτέλεσης στο τέλος του βρόχου.

Παράδειγμα

Θέλουμε να τυπώσουμε τις τετραγωνικές ρίζες των πρώτων 10 αριθμών εισόδου, αγνοώντας τους αρνητικούς.

```
for (int i{0}; i < 10; ++i) {
   double x;
   std::cin >> x;
   if (x < 0.0) {
     continue;
   }
   std::cout << u8"Η τετραγωνική ρίζα είναι "
             << std::sqrt(x) << '\n';
}
```