

## ΑΣΚΗΣΕΙΣ ΕΠΑΝΑΛΗΨΗΣ 2020 – Σύνολο 2<sup>ο</sup>

### ΑΣΚΗΣΗ 1

Δίνεται το παρακάτω πρόγραμμα σε μια γλώσσα τύπου Pascal στην οποία ισχύει ο στατικός κανόνας εμβέλειας:

```
program MAIN;
  var i, a: integer;
  function F(j, b, c, d: integer): integer;
    var e: integer;
    begin
      e:= 1;
      for j:= b to c do e:= e*d;
      F:= e
    end;
BEGIN
  i:= 1;
  e:= i;
  a:= F(i, 1, 10, i);
  write(a)
END.
```

- Ποια είναι τα περιβάλλοντα αναφοράς (τοπικά, μη-τοπικά, καθολικά) όλων των τμημάτων του προγράμματος;
- Στον παραπάνω κώδικα υπάρχει μία εντολή η οποία δεν είναι σωστή από σημασιολογική άποψη και θα προκαλέσει run-time error. Ποια είναι αυτή και γιατί; «Σβήστε» την εντολή αυτή από τον κώδικα.
- Τι υπολογίζει η συνάρτηση F και τυπώνεται στο τέλος με την εντολή write(a), στις παρακάτω περιπτώσεις τρόπου μεταβίβασης παραμέτρων (παρουσιάστε και εξηγήστε τις αλλαγές τιμών όλων των μεταβλητών κατά τη διαδικασία υπολογισμού);
  - Κλήση με τιμή (call by value)
  - Κλήση με αναφορά (call by reference)

Δώστε το αποτέλεσμα σας είτε με αριθμό, είτε με μαθηματικό τύπο.

## ΑΣΚΗΣΗ 2

Δίνεται το παρακάτω πρόγραμμα σε μια γλώσσα τύπου Pascal στην οποία ισχύει ο στατικός κανόνας εμφάνισης:

```
program MAIN;
  var z: integer;
      a: array [1..2] of integer;
  procedure P(x: integer);
    begin
      a[1]:= 6;
      z:= 2;
      x:= x + 5
    end;
BEGIN
  a[1]:= 2;  a[2]:=3;
  z:= 1;  x:= 2;
  P(a[z]);
  write(a[1], a[2], z)
END.
```

- a) Ποια είναι τα περιβάλλοντα αναφοράς (τοπικά, μη-τοπικά, καθολικά) όλων των τμημάτων του προγράμματος;
- b) Στον παραπάνω κώδικα υπάρχει/ουν εντολή/ές που δεν είναι σωστή/ές από σημασιολογική άποψη και θα προκαλέσει/ουν run-time error. Ποια/ες είναι αυτή/ες και γιατί; «Σβήστε» την/τις εντολή/ες από τον κώδικα.
- c) Τι τυπώνεται στο τέλος με την εντολή **write**, στις παρακάτω περιπτώσεις τρόπου μεταβίβασης παραμέτρων (παρουσιάστε και εξηγήστε τις αλλαγές τιμών όλων των μεταβλητών κατά τη διαδικασία υπολογισμού);
  - i. Κλήση με τιμή (call by value)
  - ii. Κλήση με τιμή – αποτέλεσμα (call by value – result)
  - iii. Κλήση με αναφορά (call by reference)
  - iv. Κλήση με όνομα (call by name)

### ΑΣΚΗΣΗ 3

Δίνεται το παρακάτω τμήμα προγράμματος C++:

1	<code>int main()</code>
2	<code>{</code>
3	<code>int numbers[5];</code>
4	<code>int *p;</code>
5	<code>p = numbers; *p = 10;</code>
6	<code>p = numbers + 3; *p = 40;</code>
7	<code>p = &amp;numbers[2]; *p = 30;</code>
8	<code>p = numbers; *(p+4) = 50;</code>
9	<code>p++; *p = 20;</code>
10	<code>for (int n=0; n&lt;5; n++)</code>
11	<code>cout &lt;&lt; numbers[n] &lt;&lt; ", ";</code>
12	<code>return 0;</code>
13	<code>}</code>

- a) Εξηγήστε γραμμή – γραμμή τους ορισμούς και τη λειτουργία του παραπάνω προγράμματος.  
b) Τι θα τυπωθεί με την εκτέλεση του προγράμματος;

### ΑΣΚΗΣΗ 4

- a)
- Δίνεται η εντολή της C: `if A < B || C < D then ...` Δείξτε τη σειρά εκτέλεσης των υπολογισμών χρησιμοποιώντας παρενθέσεις.
  - Κάντε το ίδιο για την αντίστοιχη εντολή στην Pascal: `if A < B or C < D then ...`
  - Για τις δύο παραπάνω περιπτώσεις, υπάρχουν κάποιες προϋποθέσεις για τους τύπους των μεταβλητών **A, B, C, D**, ώστε να μην επιστραφεί λάθος κατά τη μετάφραση;
- b) Στην παρακάτω έκφραση της C, χρησιμοποιούνται ακέραιες μεταβλητές με τρέχουσες τιμές: A = 1, B = 2, C = 3, D = 4, E = 5:

$$A \ || \ B < C \ \&\& \ D + (E = 1)$$

Ποια τιμή έχει η έκφραση; Αριθμήστε τη χρονική ακολουθία των πράξεων στην έκφραση, χρησιμοποιώντας και παρενθέσεις.