

ΕΞΑΣΦΑΛΙΣΗ ΠΟΙΟΤΗΤΑΣ ΚΑΙ ΠΡΟΤΥΠΑ

ΕΡΓΑΣΙΑ 3

1. Α' υλοποίηση

```
void sort_numbers_ascending(int number[], int count)
```

| Τελεστές | Αριθμός εμφανίσεων | Έντελα | Αριθμός εμφανίσεων |
|-----------------------------------|---------------------|---------------------------------|---------------------|
| void | 1 | count | 4 |
| sort_numbers_ascending(,) { } * | 1 | temp | 3 |
| int | 3 | i | 5 |
| for(; ;) ** | 3 | j | 8 |
| if() *** | 1 | k | 7 |
| printf() **** | 2 | "Numbers in ascending order:\n" | 1 |
| < | 3 | "%d\n" | 1 |
| > | 1 | 0 | 2 |
| = | 6 | 1 | 1 |
| ++ | 3 | | |
| + | 1 | | |
| number[] | 8 | | |
| , | 4 | | |
| ; | 6 | | |
| { } | 3 | | |
| | | | |
| n ₁ = 15 | N ₁ = 46 | n ₂ = 9 | N ₂ = 32 |

Line of Code = 19

Line of Comments = 0

* Η συγκεκριμένη ρουτίνα έχει υποχρεωτικά ένα ζεύγος από παρενθέσεις ένα κόμμα και ένα ζεύγος αγκίστρων.

** Η for έχει υποχρεωτικά ένα ζεύγος από παρενθέσεις και δύο ερωτηματικά.

*** Η if έχει υποχρεωτικά ένα ζεύγος από παρενθέσεις.

**** Η printf έχει υποχρεωτικά ένα ζεύγος από παρενθέσεις.

void main()

| Τελεστές | Αριθμός εμφανίσεων | Έντελα | Αριθμός εμφανίσεων |
|---------------------------------|---------------------|--|---------------------|
| void | 1 | i | 5 |
| main() {} * | 1 | count | 3 |
| int | 1 | 20 | 2 |
| number[] | 1 | t | 2 |
| printf() | 3 | 0 | 2 |
| scanf(,) * * | 3 | "How many numbers you are going to enter:" | 1 |
| while() *** | 1 | "%d" | 3 |
| for(; ;) | 1 | &count | 2 |
| sort_number_ascending(,) **** | 1 | "\nEnter the numbers one by one:" | 1 |
| < | 1 | "\nThis is a test" | 1 |
| > | 1 | &number[] | 1 |
| = | 2 | number | 1 |
| ++ | 1 | | |
| , | 3 | | |
| ; | 8 | | |
| {} | 1 | | |
| | | | |
| n ₁ = 16 | N ₁ = 30 | n ₂ = 12 | N ₂ = 24 |

Line of Comments = 1

Line of Code = 16

* Η ρουτίνα main έχει υποχρεωτικά ένα ζεύγος από παρενθέσεις και ένα ζεύγος αγκίστρων.

** Η scanf έχει υποχρεωτικά ένα ζεύγος από παρενθέσεις και ένα κόμμα.

*** Η while έχει υποχρεωτικά ένα ζεύγος από παρενθέσεις.

**** Όταν καλούμε την συγκεκριμένη ρουτίνα χρειάζονται υποχρεωτικά ένα ζεύγος παρενθέσεων και ένα κόμμα (χρειάζονται 2 παράμετροι).

Β' υλοποίηση

```
void main() {}
```

| Τελεστές | Αριθμός εμφανίσεων | Έντελα | Αριθμός εμφανίσεων |
|---------------------|---------------------|--|---------------------|
| void | 1 | i | 16 |
| main() {} | 1 | 20 | 4 |
| int | 2 | t | 4 |
| num[] | 9 | 0 | 4 |
| printf() | 6 | n | 2 |
| scanf(,) | 5 | count | 3 |
| while() | 1 | j | 7 |
| for(; ;) | 5 | a | 2 |
| < | 5 | x | 1 |
| > | 2 | b | 1 |
| = | 9 | "How many numbers you are going to enter:" | 1 |
| ++ | 4 | "%d" | 5 |
| + | 1 | "\nEnter the numbers one by one:" | 1 |
| , | 8 | "\nThis is a test" | 1 |
| ; | 16 | &count | 4 |
| {} | 5 | "\nThis is my test" | 1 |
| -- | 1 | &num[] | 1 |
| | | "Numbers in ascending order:\n" | 1 |
| | | "%d\n" | 1 |
| | | 1 | 1 |
| | | | |
| | | | |
| n ₁ = 17 | N ₁ = 81 | n ₂ = 20 | N ₂ = 61 |

Line of Comments = 12

Line of Code = 45

2. Α' Υλοποίηση

```
void sort_numbers_ascending
```

$$\text{Nest}/N = n_1 * \log_2(n_1) + n_2 * \log_2(n_2) / N_1 + N_2 = 15 * \log_2(15) + 9 * \log_2(9) / 46 + 32 \approx 1.117$$

$$L = V^* / V \approx L_{\text{est}} = 2 * n_2 / n_1 * N_2 = 2 * 9 / 15 * 32 \approx 0.038$$

$$\lambda = L * V^* \approx L^2 * V = L^2 * N * \log_2(n) = L^2 * (N1 + N2) * \log_2(n1 + n2) = (0.038)^2 * (46 + 32) * \log_2(15 + 9) \approx 0.503$$

Lines of Comments / Physical Lines of Code = 0 / 19 = 0

A' Υλοποίηση

void main

$$Nest/N = n1 * \log_2(n1) + n2 * \log_2(n2) / N1 + N2 = 16 * \log_2(16) + 12 * \log_2(12) / 30 + 24 \approx 1.982$$

$$L = V^* / V \approx L_{est} = 2 * n2 / n1 * N2 = 2 * 12 / 16 * 24 \approx 0.063$$

$$\lambda = L * V^* \approx L^2 * V = L^2 * N * \log_2(n) = L^2 * (N1 + N2) * \log_2(n1 + n2) = (0.063)^2 * (30 + 24) * \log_2(16 + 24) \approx 1.014$$

Lines of Comments / Physical Lines of Code = 1 / 16 \approx 0.063

B' Υλοποίηση

void main

$$Nest/N = n1 * \log_2(n1) + n2 * \log_2(n2) / N1 + N2 = 17 * \log_2(17) + 20 * \log_2(20) / 81 + 61 \approx 1.098$$

$$L = V^* / V \approx L_{est} = 2 * n2 / n1 * N2 = 2 * 20 / 17 * 61 \approx 0.039$$

$$\lambda = L * V^* \approx L^2 * V = L^2 * N * \log_2(n) = (0.039)^2 * (61 + 81) * \log_2(17 + 20) \approx 1.101$$

Lines of Comments / Physical Lines of Code = 12 / 45 \approx 0.267

Σχόλια:

1. Το V^* δεν μπορεί να υπολογιστεί με πραγματικά δεδομένα, οπότε για τον υπολογισμό του, χρησιμοποιούνται προσεγγίσεις.
2. Τα αριθμητικά αποτελέσματα είναι στρογγυλοποιημένα στο τρίτο δεκαδικό ψηφίο.
3. Για τον υπολογισμό των physical lines of codes, μετρήθηκαν και οι κενές γραμμές.

3.

Σ1.:

$$\text{Nest} / N = 1.117 + 1.982 / 2 \approx 1.5495$$

$$L = V^* / V = 0.038 + 0.063 / 2 \approx 0.0505$$

$$\lambda = 0.503 + 1.014 / 2 \approx 0.759$$

$$\text{Lines of Comments} / \text{Physical Lines of Code} = 0 + 0.063 / 2 \approx 0.032$$

Σ2.:

$$N(\text{sort}) = N1 + N2 = 46 + 32 = 78$$

$$N(\text{main}) = N1 + N2 = 30 + 24 = 54$$

$$N(\text{total}) = N(\text{sort}) + N(\text{main}) = 132$$

$$\text{Nest} / N = 1.117 * 78 + 1.982 * 54 / 132 \approx 1.471$$

$$L = V^* / V = 0.038 * 78 + 0.063 * 54 / 132 \approx 0.048$$

$$\lambda = 0.503 * 78 + 1.014 * 54 / 132 \approx 0.712$$

$$\text{Lines of Comments} / \text{Physical Lines of Code} = 0 * 78 + 0.063 * 54 / 132 \approx 0.026$$

Για τον υπολογισμό του μέσου όρου, και οι δύο όροι επηρεάζουν το τελικό αποτέλεσμα με την ίδια ισχύ. Αντιθέτως, για τον υπολογισμό του σταθμισμένου μέσου όρου, επειδή χρησιμοποιούμε το κριτήριο της στάθμισης έχουμε ανομοιόμορφη επιρροή των όρων στο τελικό αποτέλεσμα. Στην περίπτωση που έχουμε πολλές ρουτίνες, παρατηρούμε ότι η επιλογή του μέσου όρου για τον υπολογισμό μιας μετρικής οδηγεί σε λάθος αποτελέσματα λόγω της μεγάλης επιρροής της εκτενούς ρουτίνας στην τελική μέτρηση. Αντίθετα, η επιλογή του σταθμικού μέσου όρου για τον υπολογισμό μιας μετρικής οδηγεί σε πιο σωστά αποτελέσματα λόγω της ύπαρξης ενός γενικότερου κριτηρίου στάθμισης.

4.

| Μετρήσεις | A' Υλοποίηση | B' Υλοποίηση |
|---|--------------|--------------|
| Nest / N | 1.471 | 1.098 |
| L | 0.048 | 0.039 |
| λ | 0.712 | 1.101 |
| Lines of Comments / Physical Lines of Code | 0.026 | 0.267 |

Παρατηρούμε ότι η A' Υλοποίηση έχει λίγο μεγαλύτερο μήκος προγράμματος, μεγαλύτερο επίπεδο προγράμματος, μικρότερο επίπεδο γλώσσας και μικρότερο ποσοστό σχολίων σε σχέση με την B' Υλοποίηση. Θεωρώ πως η καλύτερη υλοποίηση είναι η A' γιατί παρόλο που έχει αρκετά μεγαλύτερο μήκος προγράμματος είναι πιο κατανοητή αφού έχει μεγαλύτερο L.