

3^η Εργασία

1.

Υλοποίηση Α

Τελεστές	Αριθμός εμφανίσεων	Έντελα	Αριθμός εμφανίσεων
void	1	fib1	4
main()	1	fib2	5
int	1	fib3	4
=	7	count	4
,	9	N	4
;	12	0	2
printf()	5	1	1
scanf()	1	"Enter the value of N: "	1
&	1	"%d"	1
while()	1	"First %d FIBONACCI numbers are...\n"	1
<	1	"%d\n"	3
{}	2	2	1
+	1		
++	1		
n ₁ = 14	N ₁ = 44	n ₂ = 12	N ₂ = 31

Υλοποίηση Β

Τελεστές	Αριθμός εμφανίσεων	Έντελα	Αριθμός εμφανίσεων
void	1	i	5
main()	1	fib1	4
int	1	fib2	5
=	6	fib3	4
,	9	N	4
;	10	0	1
printf()	5	1	1
scanf()	1	"Enter the value of N: "	1
&	1	"%d"	1
for(; ;)	1	"First %d FIBONACCI numbers are...\n"	1
<	1	"%d\n"	3
++	1	2	2
{}	3		
+	1		
if()	1		
<=	1		
n ₁ = 16	N ₁ = 44	n ₂ = 12	N ₂ = 32

Σημείωση: τα σχόλια δεν τα έχω λάβει υπόψη μου, διότι είναι κομμάτια κώδικα που δεν θα εκτελεστούν. Η for(; ;) τοποθετήθηκε έτσι όπως είναι στις διαφάνειες και επειδή όπως είπαμε και στο μάθημα, οι παρενθέσεις και τα 2 “;” υπάρχουν πάντα. Επίσης η main() είναι ξεχωριστός τελεστής από το void, καθώς η main() δεν συνοδεύεται πάντα από void. Ακόμη το “&” θεωρήθηκε ξεχωριστός τελεστής διότι δεν συνοδεύεται απαραίτητα με την μεταβλητή N. Τέλος τα “{}” θεωρήθηκαν ξεχωριστός τελεστής σε κάθε συνάρτηση, καθώς σε όλες τις περιπτώσεις η συνάρτηση μπορούσε να υπάρξει και χωρίς αυτά.

2.

Υλοποίηση A

- **λόγος του εκτιμητή μήκους προς το μήκος προγράμματος (Nest/N)**

$$N_{est} = n_1 \log_2 n_1 + n_2 \log_2 n_2 = 3,807 * 14 + 3,584 * 12 = 53,302 + 43,019 = 96,322$$

$$N = N_1 + N_2 = 75$$

$$N_{est}/N = 1,284$$

- **επίπεδο προγράμματος (L)**

$$L = V^*/V = (2 * n_2)/(n_1 * N_2) = 24/(14 * 31) = 24/434 = 0,055$$

- **επίπεδο γλώσσας του (λ)**

$$V = (N_1 + N_2) * \log_2(n_1 + n_2) = 75 * 4,7 = 352,532$$

$$\lambda = LV^* = L^2 V = 0,003 * 352,532 = 1,078$$

- **λόγος αριθμού γραμμών σχολίων προς τον αριθμό φυσικών γραμμών κώδικα (Lines of Comments / Physical Lines of Code)**

Η υλοποίηση A περιέχει 4 γραμμές σχολίων και 26 φυσικές γραμμές κώδικα. Άρα Lines of Comments / Physical Lines of Code = 4/26 = 0,153 (κατά προσέγγιση) = 15,3%.

Υλοποίηση B

- **λόγος του εκτιμητή μήκους προς το μήκος προγράμματος (Nest/N)**

$$N_{est} = n_1 \log_2 n_1 + n_2 \log_2 n_2 = 4 * 16 + 3,584 * 12 = 64 + 43,019 = 107,019$$

$$N = N_1 + N_2 = 76$$

$$N_{est}/N = 1,408$$

- **επίπεδο προγράμματος (L)**

$$L = V^*/V = (2 * n_2)/(n_1 * N_2) = 24/512 = 0,046$$

- **επίπεδο γλώσσας του (λ)**

$$V = (N_1 + N_2) * \log_2(n_1 + n_2) = 76 * 4,807 = 365,358$$

$$\lambda = LV^* = L^2 V = 0,002 * 365,358 = 0,802$$

• **λόγος αριθμού γραμμών σχολίων προς τον αριθμό φυσικών γραμμών κώδικα (Lines of Comments / Physical Lines of Code)**

Η υλοποίηση B περιέχει 19 γραμμές σχολίων και 46 φυσικές γραμμές κώδικα. Άρα $\text{Lines of Comments / Physical Lines of Code} = 19/46 = 0.413$ (κατά προσέγγιση) = 41,3%.

Σημείωση: σε όλα τα παραπάνω έχει γίνει στρογγυλοποίηση στα 3 δεκαδικά ψηφία.

3. Παρουσιάζονται συγκεντρωμένες σε έναν πίνακα οι τιμές των μετρικών για τις 2 υλοποιήσεις, για να βοηθήσουν στην εξαγωγή συμπερασμάτων.

	Nest/N	L	λ	Lines of Comments/P-LOC
A Υλοποίηση	1,284	0,055	1,078	15,3%
B Υλοποίηση	1,408	0,046	0,802	41,3%

Παρατηρώ ότι η A Υλοποίηση έχει καλύτερα αποτελέσματα στο επίπεδο προγράμματος (L) και στο επίπεδο γλώσσας (λ), ενώ η B Υλοποίηση στο λόγο αριθμού γραμμών σχολίων προς τον αριθμό φυσικών γραμμών κώδικα, και στον λόγο του εκτιμητή μήκους προς το μήκος προγράμματος (Nest/N) (το Nest μας δείχνει ότι υπάρχει εξάρτηση του μήκους του προγράμματος μόνο από τους διακριτούς τελεστές και έντελα). Μια ακόμα παρατήρηση είναι, ότι το ποσοστό σχολίων στην B Υλοποίηση είναι πολύ μεγαλύτερο (σχεδόν τριπλάσιο) από ότι στην A Υλοποίηση. Ωστόσο, τα περισσότερα σχόλια στην B Υλοποίηση δεν επεξηγούν την λειτουργία του κώδικα, ένα σχόλιο αφορά προφανώς μια παλαιότερη έκδοση του κώδικα, ενώ υπήρχαν και κάποια τα οποία μπορεί να θεωρηθούν και περιττά (όπως το «this is my code» και τα πολλά κενά). Αντίθετα στην A Υλοποίηση παρόλο που υπάρχουν πολύ λίγα σχόλια, αυτά είναι αρκετά περιγραφικά και επεξηγούν την λειτουργία του κώδικα. Επιπρόσθετα, όπως αναφέρεται και στο pdf που υπάρχει στο eclass «Βοηθητικό για Halstead metrics» το Nest αρκετές φορές έχει μεγάλα ποσοστά λάθους. Από τα παραπάνω, συμπεραίνω ότι καλύτερη υλοποίηση φαίνεται πως είναι η A, καθώς έχει υψηλότερο επίπεδο προγράμματος, υψηλότερο επίπεδο γλώσσας, -άρα είναι πιο εύκολο στην χρήση και στην συντήρησή του- και έχει πιο επεξηγηματικά σχόλια (παρόλο που είναι λίγα).