

Περιεχόμενα

Ζητούμενο 1	2
Υλοποίηση A	2
Υλοποίηση B	3
Ζητούμενο 2	4
Υλοποίηση A	4
Λόγος του εκτιμητή μήκους προς το μήκος προγράμματος	4
Επίπεδο προγράμματος	4
Επίπεδο γλώσσας	4
Λόγος αριθμού γραμμών σχολίων προς τον αριθμό φυσικών γραμμών κώδικα	4
Υλοποίηση B	4
Λόγος του εκτιμητή μήκους προς το μήκος προγράμματος	4
Επίπεδο προγράμματος	4
Επίπεδο γλώσσας	4
Λόγος αριθμού γραμμών σχολίων προς τον αριθμό φυσικών γραμμών κώδικα	4
Ζητούμενο 3	5

Ζητούμενο 1

Υλοποίηση Α

- Τα σχόλια δεν προσμετρήθηκαν, καθώς δεν προσθέτουν στη λειτουργικότητα του κώδικα και δεν επηρεάζουν την υλοποίηση.
- Κάθε συνάρτηση μετρίεται με τις “()” (πάντα τις χρειαζόμαστε μαζί), αλλά όχι με τα “{}”, καθώς εάν την καλέσουμε τα “{}” δεν είναι απαραίτητα.
- Η scanf() θεωρούσα ότι ορίζεται έτσι “scanf(,)”, αφού έχει πάντα “,”, όμως σύμφωνα με τις διαφάνειες, έχει ξεκάθαρα οριστεί ως ο τελεστής “scanf()”. Άρα και εγώ χρησιμοποίησα αυτό.

Αγνοώντας τα #include, ο πίνακας για την Υλοποίηση Α έχει ως εξής:

Τελεστές	Αριθμός εμφανίσεων	Εντελα	Αριθμός εμφανίσεων
void	1	fib1	4
main()	1	0	2
{}	2	fib2	5
int	1	1	1
=	7	fib3	4
,	9	Count	4
;	12	2	1
printf()	5	N	4
scanf()	1	"Enter the value of N: "	1
&	1	"%d"	1
while()	1	"First %d FIBONACCI numbers are...\n"	1
<	1	"%d\n"	3
+	1		
++	1		
n1=14	N1=44	n2=12	N2=31

Υλοποίηση Β

- Η for() ορίζεται ως ο τελεστής “for(;;)”, καθώς υποχρεωτικά θα έχει “;;”.

Ο πίνακας για την Υλοποίηση Β, με την ίδια λογική που χρησιμοποιήθηκε στην προηγούμενη υλοποίηση, έχει ως εξής:

Τελεστές	Αριθμός εμφανίσεων	Εντελα	Αριθμός εμφανίσεων
void	1	l	5
main()	1	fib1	4
{}	3	0	1
int	1	fib2	4
=	6	1	1
,	9	fib3	5
;	10	N	4
printf()	5	"Enter the value of N: "	1
scanf()	1	"%d"	1
&	1	"First %d FIBONACCI numbers are...\n"	1
for(;;)	1	"%d\n"	3
if()	1	2	2
+	1		
++	1		
<	1		
<=	1		
n1=16	N1=44	n2=12	N2=32

Ζητούμενο 2

Υλοποίηση Α

Λόγος του εκτιμητή μήκους προς το μήκος προγράμματος

Έχουμε ότι $N_{est} = n_1 \log_2 n_1 + n_2 \log_2 n_2 = 53.3029689088 + 43.0195500087 = 96.3225189175$.

Επίσης $N = N_1 + N_2 = 75$.

Άρα ο λόγος του εκτιμητή μήκους προς το μήκος προγράμματος θα είναι $N_{est}/N = 1.28430025223$.

Επίπεδο προγράμματος

Το επίπεδο προγράμματος θα είναι $L_{est} = 2n_2/n_1N_2 = 24 / 434 = 0.05529953917$.

Χρησιμοποιήθηκε το " L_{est} ", καθώς δεν είχαμε το V^* ώστε μπορούμε να υπολογίσουμε ακριβώς το " L ".

Επίπεδο γλώσσας

Έχουμε ότι ο όγκος θα είναι $V = N \log_2 n = 75 * 4.700439718141093 = 352.532978861$.

Επίσης, θα χρησιμοποιήσουμε ως L το L_{est} , καθώς με βάση τα στοιχεία είναι το μόνο που μπορούμε να βρούμε ως εκτίμηση του L .

Άρα το επίπεδο γλώσσας θα είναι $\lambda = LV^* = L^2V = 0.00305803903 * 352.532978861 = 1.07805960872$.

Λόγος αριθμού γραμμών σχολίων προς τον αριθμό φυσικών γραμμών κώδικα

Lines of Comments / Physical Lines of Code = 4 / 26

Οι γραμμές μετρήθηκαν κατευθείαν από το αρχείο της εκφώνησης, μαζί με τα κενά, ως Physical Lines of Code.

Υλοποίηση Β

Όλες οι μετρήσεις έγιναν ομοίως με τις μετρήσεις της προηγούμενης υλοποίησης.

Λόγος του εκτιμητή μήκους προς το μήκος προγράμματος

Έχουμε ότι $N_{est} = n_1 \log_2 n_1 + n_2 \log_2 n_2 = 64 + 43.0195500087 = 107.019550009$.

Επίσης $N = N_1 + N_2 = 76$.

Άρα ο λόγος του εκτιμητή μήκους προς το μήκος προγράμματος θα είναι $N_{est}/N = 1.4081519738$.

Επίπεδο προγράμματος

Το επίπεδο προγράμματος θα είναι $L_{est} = 2n_2/n_1N_2 = 24 / 512 = 0.046875$.

Επίπεδο γλώσσας

Έχουμε ότι ο όγκος θα είναι $V = N \log_2 n = 365.358974076$.

Άρα το επίπεδο γλώσσας θα είναι $\lambda = LV^* = L^2V = 0.00219726562 * 365.358974076 = 0.80279071269$.

Λόγος αριθμού γραμμών σχολίων προς τον αριθμό φυσικών γραμμών κώδικα

Lines of Comments / Physical Lines of Code = 19 / 46

Ζητούμενο 3

Καταλληλότερη θεωρώ την υλοποίηση A, καθώς υλοποιείται σε λιγότερες γραμμές (26 έναντι 46 του B), είναι πιο εύκολη στην κατανόηση και παράλληλα αν και έχει λίγα σχόλια (4 γραμμές έναντι 19 του B), τα σχόλια αυτά είναι όσο επεξηγηματικά χρειάζεται για να μπορεί κάποιος να καταλάβει κατευθείαν πως λειτουργεί ο κώδικας. Αντίθετα, στην υλοποίηση B, τα σχόλια είναι πολλά και δεν είναι τόσο ξεκάθαρα. Αυτό έχει ως αποτέλεσμα η A να είναι πιο εύκολο να αναλυθεί και να επεκταθεί, πράγμα συχνά πολύ σημαντικό. Επίσης, η υλοποίηση B για $N = 2$ δε λειτουργεί, καθώς δεν εμφανίζει τους 2 πρώτους αριθμούς Fibonacci, ενώ η A λειτουργεί κανονικά για κάθε είσοδο.

Σύμφωνα με τις μετρικές, η A είναι καλύτερη από την B, αφού έχει υψηλότερο επίπεδο προγράμματος L, (0.05529953917 έναντι 0.046875 της B) και υψηλότερο επίπεδο γλώσσας λ (1.07805960872 έναντι 0.80279071269 της B). Δηλαδή, το A είναι καλύτερα και πιο αποδοτικά γραμμένο στη συγκεκριμένη γλώσσα, πράγμα εξίσου σημαντικό.

Συμπερασματικά, το σενάριο της υλοποίησης A είναι πάντα προτιμότερο, αφού είναι συνοπτικότερα γραμμένο, έχει πιο ξεκάθαρα σχόλια, είναι πιο κατανοητό στη λειτουργία του και στην ανάλυση του, είναι ευκολότερο να επεκταθεί, είναι γραμμένο πιο «κοντά» στους κανόνες της γλώσσας και δουλεύει σωστά για κάθε είσοδο.