ΕΡΩΤΗΜΑ 1

- Παραδοχές

1. Στον ορισμό της συνάρτησης sort_numbers_ascending, έχω θεωρήσει το void αλλά και το όνομα της συνάρτησης ως τελεστές. Το void ουσιαστικά έχει τη δράση ότι κάνει τη συνάρτηση να μην επιστρέφει κάτι και η συνάρτηση δρα πάνω στα ορίσματα ώστε να κάνει έναν υπολογισμό. Το είχα σκεφτεί βέβαια να το βάλω ως έντελο, λόγω του ότι το void είναι τελεστής που δρα πάνω στη συνάρτηση, ωστόσο, σκέφτηκα ότι ως έντελο θα μπορούσα να το θεωρήσω περισσότερο αν είχα δήλωση συνάρτησης και όχι τώρα που έχω ορισμό.

2. Για το number[] και το num[] ουσιαστικά ακολούθησα τη λογική που είχαμε δει στις διαφάνειες, δηλαδή θεώρησα ως τελεστές τους πίνακες και ως έντελα τα στοιχεία των πινάκων ώστε να είναι εμφανές ότι γίνεται μία πράξη.

3. Όταν μετράω τις γραμμές κώδικα και σχολίων στην Α' υλοποίηση, θεωρώ ότι το include είναι γραμμή κώδικα και για τις δύο ρουτίνες. Αυτό συμβαίνει γιατί θεωρώντας υποθετικά ότι χωρίζαμε τον κώδικα σε δύο αρχεία, θα βάζαμε και στα δύο include. Με τα σχόλια, έχω βάλει τα πρώτα που εξηγούν τη λειτουργία του προγράμματος, στη 2η ρουτίνα γιατί θα τα βάζαμε στο αρχείο με τη main.

4. Στον υπολογισμό του L, χρησιμοποίησα το Lest όπως είχε ειπωθεί.

Α' υλοποίηση, 1η ρουτίνα

Τελεστές	Αριθμός εμφανίσεων	Έντελα	Αριθμός εμφανίσεων
void	1	count	4
sort_numbers_ascending()	1	temp	3
int	3	i	5
number[]	8	j	8
for(; ;)	3	k	7
,	5	"Numbers in ascending order:\n"	1
=	6	"%d\n"	1
<	3	0	2
++	3	1	1
+	1		
if( )	1		
>	1		
printf( )	2		
;	6		
{}	3		
n_1=15	N_1=47	n_2=9	N_2=32

Α' υλοποίηση, 2η ρουτίνα

Τελεστές	Αριθμός εμφανίσεων	Έντελα	Αριθμός εμφανίσεων
void	1	i	5
main()	1	count	5
int	1	20	2
number[]	2	t	2
,	6	0	2
=	2	"How many numbers you are going to enter:"	1
printf()	3	"%d"	3
scanf()	3	"\nEnter the numbers one by one:"	1
&	3	"\nThis is a test"	1
while()	1	number	1
>	1		
for(; ;)	1		
<	1		
++	1		
sort_numbers_ascending(,)	1		
;	8		
{}	2		
n_1=17	N_1=38	n_2=10	N_2=23

Β' υλοποίηση

Τελεστές	Αριθμός εμφανίσεων	Έντελα	Αριθμός εμφανίσεων
void	1	i	16
main()	1	20	4
int	2	t	5
num[]	9	0	4
=	9	n	3
printf()	6	count	7
scanf()	5	j	7
&	5	a	3
while()	1	x	1
>	2	b	1
for(; ;)	5	"How many numbers you are going to enter:"	1
<	5	"%d"	5

++	4	“\nEnter the numbers one by one:”	1
,	13	“\nThis is a test”	1
;	16	“\nThis is my test”	1
{}	6	“Numbers in ascending order:\n”	1
--	1	1	1
+	1	“%d\n”	1
If()	1		
n_1=19	N_1=93	n_2=18	N_2=63

ΕΡΩΤΗΜΑ 2

Α’ υλοποίηση, 1η ρουτίνα

Λεξιλόγιο: **n=**n_1+n_2=15+9=**24**

Μήκος προγράμματος: **N=**N1+N2=47+32=**79**

Εκτιμητής μήκους: **Nest=** $n_1\log_2 n_1 + n_2\log_2 n_2$ = 15*3.9 +9*3.2=**87.3**

Όγκος: **V=** $N\log_2 n$ =79*4.6=**363.4**

Lines of comments: 1

Physical lines of code: 23

- λόγος του εκτιμητή μήκους προς το μήκος προγράμματος του Halstead: **Nest/N=1.1**
- επίπεδο προγράμματος του Halstead $L = L_{est} = \frac{2 \cdot n_2}{n_1 \cdot N_2} = 0.037$
- επίπεδο γλώσσας του Halstead **λ=**L^2*V=0.5
- λόγος αριθμού γραμμών σχολίων προς τον αριθμό φυσικών γραμμών κώδικα (Lines of Comments / Physical Lines of Code): 1/23=0.04

Α’ υλοποίηση, 2η ρουτίνα

Λεξιλόγιο: **n=**n_1+n_2=17+10**=27**

Μήκος προγράμματος: **N=**N1+N2=38+23**=61**

Εκτιμητής μήκους: **Nest=** $n_1\log_2 n_1 + n_2\log_2 n_2$ = 17*4.08 + 10*3.32 **= 102.56**

Όγκος: **V=** $N\log_2 n$ =61*4.75**=289.75**

Lines of comments: 5

Physical lines of code: 23

- λόγος του εκτιμητή μήκους προς το μήκος προγράμματος του Halstead: **Nest/N**=102.56/61=**1.68**
- επίπεδο προγράμματος του Halstead $L = L_{est} = \frac{2 \cdot n_2}{n_1 \cdot N_2} = 0.05$
- επίπεδο γλώσσας του Halstead **λ**=L^2*V=0.72
- λόγος αριθμού γραμμών σχολίων προς τον αριθμό φυσικών γραμμών κώδικα (Lines of Comments / Physical Lines of Code): 5/23=0.2

Β' υλοποίηση

Λεξιλόγιο: **n**=n_1+n_2=19+18=37

Μήκος προγράμματος: **N**=N1+N2=93+63=156

Εκτιμητής μήκους: **Nest=** $n_1 \log_2 n_1 + n_2 \log_2 n_2$ = 19*4.25+18*4.17 = 155.81

Όγκος: **V=** $N \log_2 n$ =156*5.2 = 811.2

Lines of comments: 19

Physical lines of code: 53

- λόγος του εκτιμητή μήκους προς το μήκος προγράμματος του Halstead: **Nest/N**=155.81/156=**0.99**
- επίπεδο προγράμματος του Halstead

$$L = L_{est} = \frac{2 \cdot n_2}{n_1 \cdot N_2} = 0.03$$

- επίπεδο γλώσσας του Halstead **λ=** L^2*V=0.73
- λόγος αριθμού γραμμών σχολίων προς τον αριθμό φυσικών γραμμών κώδικα (Lines of Comments / Physical Lines of Code): 19/53=0.36

ΕΡΩΤΗΜΑ 3

Σ1.

- λόγος του εκτιμητή μήκους προς το μήκος προγράμματος του Halstead:

$$\frac{N_{est}}{N} = \frac{1.1 + 1.68}{2} = 1.39$$

- επίπεδο προγράμματος του Halstead

$\boldsymbol{L} = \frac{0.037+0.05}{2} = 0.046$

- επίπεδο γλώσσας του Halstead $\lambda = \frac{0,5+0.72}{2} = 0.61$
- λόγος αριθμού γραμμών σχολίων προς τον αριθμό φυσικών γραμμών κώδικα (Lines of Comments / Physical Lines of Code): 0.12

Σ2.

- $$\overline{\left(\frac{N_{est}}{N}\right)} = \frac{\left(\frac{N_{est}}{N}\right)_1 \cdot N_1 + \left(\frac{N_{est}}{N}\right)_2 \cdot N_2}{N_1 + N_2} = \frac{1.1 \cdot 79 + 1.68 \cdot 61}{79 + 61} = \frac{189.38}{140} = 1.35$$

- $$\bar{L} = \frac{L_1 \cdot N_1 + L_2 \cdot N_2}{N_1 + N_2} = \frac{0.037 \cdot 79 + 0.05 \cdot 61}{79 + 61} = \frac{5.973}{140} = 0.0427$$

- $$\lambda = \frac{\lambda_1 \cdot N_1 + \lambda_2 \cdot N_2}{N_1 + N_2} = \frac{0.5 \cdot 79 + 0.72 \cdot 61}{72 + 61} = \frac{83.42}{140} = 0.596$$

- $$\left(\overline{\frac{LOCom}{PLOC}}\right) = \frac{0,04 \cdot 79 + 0,2 \cdot 61}{79 + 61} = \frac{15,36}{140} = 0.11$$

➢ Σχολιασμός αποτελεσμάτων στις περιπτώσεις Σ1,Σ2

Στη συγκεκριμένη περίπτωση που έχουμε μόνο δύο ρουτίνες, τόσο ο μέσος όρος όσο και ο σταθμισμένος μέσος όρος μας δίνουν παρόμοια αποτελέσματα. Όμως αν είχαμε περισσότερες ρουτίνες, τότε θα βλέπαμε μεγαλύτερη απόκλιση στα αποτελέσματα μας. Για να κατανοήσουμε το λόγο για τον οποίο κάτι τέτοιο ισχύει αρκεί να σκεφτούμε ότι ο αριθμητικός μέσος όρος, θεωρεί όλες τις τιμές ισάξιες δηλάδη με το ίδιο βάρος, ενώ ο σταθμισμένος μέσος όρος δίνει διαφορετικά βάρη στις διαφορετικές τιμές άρα είναι πιο αξιόπιστος και έχει μεγαλύτερη ευελιξία. Στην περίπτωσή μας, φυσικά και προτιμάμε να έχουμε βάρη, γιατί θέλουμε να λάβουμε υπόψην το μήκος προγράμματος Ν της κάθε ρουτίνας για την κάθε μετρική. Για παράδειγμα αν είχαμε μία μικρή ρουτίνα και μία αρκετά εκτενή, θα θέλαμε η διαφορά αυτή να ληφθεί υπόψην στα τελικά αποτελέσματα για να έχουμε μεγαλύτερη ακρίβεια.

ΕΡΩΤΗΜΑ 4

Μετρικές	Υλοποίηση Α	Υλοποίηση Β
Nest/N	1.35	0.99
L	0.0427	0.03
Λ	0.596	0.73
LOCom/PLOC	0.11	0.36

- Nest/N

Το Nest είναι η εκτίμηση του μήκους προγράμματος λαμβάνοντας υπόψην τους διακριτούς τελεστές και έντελα, ενώ το N είναι το μήκος προγράμματος όπου για να το βρούμε προσθέτουμε το πραγματικό πλήθος εμφανίσεων των τελεστών και των εντέλων. Όταν το Nest είναι μεγαλύτερο από το N και άρα ο λόγος τους βγαίνει μικρότερος του 1, αυτό σημαίνει ότι κάποια έντελα δεν έχουν χρησιμοποιηθεί μέσα στο πρόγραμμά μας, άρα δεν έχουμε εκτιμήσει καλά το πρόβλημα που θέλαμε να λύσουμε και πιθανώς έχουμε ορίσει παραπάνω έντελα που δεν χρειαστήκαμε. Το πρόγραμμα μας είναι πιο πολύπλοκο από ότι θα έπρεπε και χρειάζεται περιττό αποθηκευτικό χώρο. Στην αντίθετη περίπτωση, που ο λόγος βγαίνει μεγαλύτερος από το 1, σημαίνει ότι όλα τα έντελα και οι τελεστές μας χρησιμοποιούνται αρκετές φορές οπότε έχουμε κάνει ένα πιο λειτουργικό πρόγραμμα και έχουμε δώσει μια καλύτερη λύση. Η υλοποίηση Α λοιπόν είναι καλύτερη από τη Β. Κάτι τέτοιο εδώ, μπορούμε να το επαληθεύσουμε και παρατηρώντας τους δύο κώδικες, για παράδειγμα, ήδη από την πρώτη ματιά παρατηρούμε ότι στην υλοποίηση Β, δε χρησιμοποιούνται όλες οι μεταβλητές.

- L=Lest

Γενικά θέλουμε το L να είναι όσο γίνεται μεγαλύτερο. Αυτό το καταλαβαίνουμε κοιτώντας τον τύπο για τη δυσκολία προγράμματος, όπου παρατηρούμε ότι οι δύο αυτές μετρικές είναι αντιστρόφως ανάλογες άρα για να έχω μικρή δυσκολία προγράμματος, πρέπει το L να έχει μεγαλύτερη τιμή. Άρα εύκολα το βλέπουμε ότι η Α υλοποίηση είναι καλύτερη.

- λ

Το λ μας δείχνει το επίπεδο γλώσσας. Όσο μικρότερο είναι, τόσο πιο πυκνογραμμένο είναι το πρόγραμμα, το οποίο πρακτικά σημαίνει ότι η υλοποίηση είναι καλύτερη δεδομένης της γλώσσας προγραμματισμού, γιατί δεν έχουμε περιττές πράξεις, αχρησιμοποιήτες μεταβλητές, ή κακογραμμένο κώδικα που θα μπορούσε να απλουστευθεί. Το λ είναι μικρότερο στην Α υλοποίηση άρα είναι καλύτερη από τη Β.

- Σχετικά με τα σχόλια, σε γενικές γραμμές, δε μπορούμε να κάνουμε έναν αντικειμενικό σχολιασμό γιατί γενικά είναι καλό το πρόγραμμά μας να περιέχει σχόλια ώστε τόσο οι άλλοι προγραμματιστές όσο και κάποιος πιθανός πελάτης ή χρήστης να κατανοούν τη λειτουργία του. Ωστόσο, είναι απαραίτητο τα σχόλια να βγάζουν νόημα και να είναι στοχευμένα εκεί που χρειάζεται. Αν δούμε τις δύο υλοποίησεις, παρατηρούμε ότι η πρώτη έχει καλύτερο σχολιασμό,

γιατί το κάθε σχόλιο είναι επεξηγηματικό, κάτι που δε συμβαίνει στη Β, όπου τα σχόλια δε σχετίζονται με το τι κάνει ο κώδικας. Οπότε η Α υλοποίηση είναι καλύτερη, αλλά αυτό κατά τη γνώμη μου δεν εξαρτάται από το νούμερο, αλλά από την ποιότητα των σχολίων.