



Αριστοτέλειο Πανεπιστήμιο
Θεσσαλονίκης



Τμήμα
Πληροφορικής

SWITCH

Εργαστήριο Λογισμικού και
Διαδραστικών Τεχνολογιών

Χρήστος Κατσάνος
Επίκουρος Καθηγητής
ckatsanos@csd.auth.gr
Γρ. 24, Εθνικής Αντιστάσεως 16,
Καλαμαριά

Διασφάλιση Ποιότητας Συστημάτων Λογισμικού

Ακαδημαϊκό Έτος: 2022 – 2023

Εργασία 2: Μετρικές ποιότητας λογισμικού & Εγκυροποίηση λογισμικού

Γενικές Οδηγίες

- Οι εργασίες κατατίθενται ηλεκτρονικά μέσω της πλατφόρμας elearning.auth.gr. Καταθέτετε ένα ενιαίο αρχείο σε κειμενογράφο της επιλογής σας ([προσοχή όχι pdf αρχείο](#)). Εργασίες που θα κατατεθούν με οποιονδήποτε άλλο τρόπο (π.χ. email) δεν λαμβάνονται υπόψη.
- Η καταληκτική ημερομηνία υποβολής της εργασίας είναι 22/01/2023. Δεν θα υπάρξει καμία παράταση στην ημερομηνία αυτή. Εργασίες που θα υποβληθούν εκπρόθεσμα δεν λαμβάνονται υπόψη.
- Σε αγκύλες καταγράφεται το ποσοστό της βαθμολογίας που αναλογεί σε κάθε μέρος ή/και υποερώτημα της εργασίας.
- Αν κρίνετε σκόπιμο, μπορείτε να κάνετε απαραίτητες παραδοχές, τις οποίες όμως πρέπει να καταγράψετε και να δικαιολογήσετε.
- Υπενθυμίζεται ότι το μάθημα περιλαμβάνει δύο βαθμολογούμενες εργασίες κάθε εξάμηνο διδασκαλίας. Ο μέσος όρος των βαθμολογημένων εργασιών σας συνεισφέρει μόνο θετικά στον τελικό βαθμό του μαθήματος από 0% έως 30% (βλέπε «Εισαγωγή.pdf»).



Αριστοτέλειο Πανεπιστήμιο
Θεσσαλονίκης



Τμήμα
Πληροφορικής

SWITCH

Εργαστήριο Λογισμικού και
Διαδραστικών Τεχνολογιών

Χρήστος Κατσάνος
Επίκουρος Καθηγητής
ckatsanos@csd.auth.gr
Γρ. 24, Εθνικής Αντιστάσεως 16,
Καλαμαριά

A. Μετρικές ποιότητας λογισμικού [50%]

Ζητήθηκε από δύο προγραμματιστές να υλοποιήσουν σε C ένα πρόγραμμα που θα τρέχει ως γραμμή εντολής και θα αντιγράφει ένα αρχείο σε ένα άλλο. Τα ονόματα των δύο αρχείων θα δίνονται ως είσοδος. Παρακάτω δίνεται ο κώδικας των δύο διαφορετικών υλοποιήσεων.

Πρόγραμμα A

```
01: #include <stdio.h>
02: #include <stdlib.h>
03:
04: int main(int argc, char *argv[])
05: {
06:     FILE *in, *out;
07:     char ch;
08:
09:     if(argc!=3) {
10:         printf("You forgot to enter a filename.\n");
11:         exit(1);
12:     }
13:
14:     if((in=fopen(argv[1], "rb")) == NULL) {
15:         printf("Cannot open input file.\n");
16:         exit(1);
17:     }
18:     if((out=fopen(argv[2], "wb")) == NULL) {
19:         printf("Cannot open output file.\n");
20:         exit(1);
21:     }
22:
23:     while(!feof(in)) {
24:         ch = getc(in);
25:         if(ferror(in)) {
26:             printf("Read Error");
27:             clearerr(in);
28:             break;
29:         } else {
30:             if(!feof(in)) putc(ch, out);
31:             if(ferror(out)) {
32:                 printf("Write Error");
33:                 clearerr(out);
34:                 break;
35:             }
36:         }
37:     }
38:     fclose(in);
39:     fclose(out);
40:
41:     return 0;
42: }
```

Πρόγραμμα B

```
01: #include <stdio.h>
02: #include <stdlib.h>
03:
```



```
04: int main(int argc, char *argv[])
05: {
06:     FILE *from, *to;
07:     char ch;
08:
09:     if(argc!=3) {
10:         printf("Usage: copy <source> <destination>\n");
11:         exit(1);
12:     }
13:
14:     /* open source file */
15:     if((from = fopen(argv[1], "rb"))==NULL) {
16:         printf("Cannot open source file.\n");
17:         exit(1);
18:     }
19:
20:     /* open destination file */
21:     if((to = fopen(argv[2], "wb"))==NULL) {
22:         printf("Cannot open destination file.\n");
23:         exit(1);
24:     }
25:
26:     /* copy the file */
27:     while(!feof(from)) {
28:         ch = fgetc(from);
29:         if(ferror(from)) {
30:             printf("Error reading source file.\n");
31:             exit(1);
32:         }
33:         if(!feof(from)) fputc(ch, to);
34:         if(ferror(to)) {
35:             printf("Error writing destination file.\n");
36:             exit(1);
37:         }
38:     }
39:
40:     if(fclose(from)==EOF) {
41:         printf("Error closing source file.\n");
42:         exit(1);
43:     }
44:
45:     if(fclose(to)==EOF) {
46:         printf("Error closing destination file.\n");
47:         exit(1);
48:     }
49:
50:     return 0;
51: }
```

Ζητούμενα:

1. Υπολογίστε και για τα δύο προγράμματα τα αποτελέσματα στον παράγοντα Nest/N και στις μετρικές L και λ του Halstead αγνοώντας κώδικα που αφορά σε δηλώσεις συναρτήσεων και μεταβλητών. Η απάντησή σας πρέπει να περιλαμβάνει για κάθε πρόγραμμα (Πρόγραμμα Α, Πρόγραμμα Β) έναν αναλυτικό πίνακα όπου θα παρουσιάζονται οι τελεστές και τα έντελα,



Αριστοτέλειο Πανεπιστήμιο
Θεσσαλονίκης



Τμήμα
Πληροφορικής

SWITCH

Εργαστήριο Λογισμικού και
Διαδραστικών Τεχνολογιών

Χρήστος Κατσάνος
Επίκουρος Καθηγητής
ckatsanos@csd.auth.gr
Γρ. 24, Εθνικής Αντιστάσεως 16,
Καλαμαριά

καθώς και το πλήθος των εμφανίσεών τους. Ο πίνακας αυτός θα πρέπει να έχει την παρακάτω μορφή:

Τελεστές	Αριθμός εμφανίσεων	Έντελα	Αριθμός εμφανίσεων

[30%]

- Υπολογίστε και για τα δύο προγράμματα τη μετρική V(G) του McCabe. [10%]
- Να συγκρίνετε τις δύο υλοποιήσεις και να σχολιάσετε το αποτέλεσμα ερμηνεύοντας τις τιμές των μετρικών που υπολογίσατε στα δύο προηγούμενα υποερωτήματα. [10%]

Β. Έλεγχος διαφανούς κουτιού και εκσφαλμάτωση [50%]

Δίνεται το παρακάτω πρόγραμμα σε γλώσσα C. Οι λειτουργικές προδιαγραφές του λογισμικού είναι οι ακόλουθες:

- Το λογισμικό δέχεται έναν μη προσημασμένο ακέραιο με το πολύ 9 ψηφία.
- Δεν επιτρέπεται κενή είσοδος, ενώ οι μόνοι επιτρεπτοί χαρακτήρες είναι τα ψηφία. Για οποιαδήποτε μη αποδεκτή είσοδο, το λογισμικό θα πρέπει να εμφανίζει ανάλογο μήνυμα.
- Αν η είσοδος είναι έγκυρη, το πρόγραμμα εμφανίζει αν ο αριθμός είναι Νάρκισσος¹ ή όχι.

```
01: #include <stdio.h>
02: #include <stdlib.h>
03: #define N 9
04:
05: int main()
06: {
07:     char input[N+3];
08:     int len,i, psifia = 0, valid;
09:     long int x, sum = 0, temp, yp, p;
10:
11:     do {
12:         printf("Dwste enan akeraio (mexri %d psifia):",N);
13:         fgets(input, sizeof input, stdin);
14:         input[strlen(input)-1]='\0';
15:         fflush(stdin);
16:         len = strlen(input);
17:         valid = 1;
18:         for (i=0;i<len;i++)
19:             if (input[i]<'0' || input[i]>'9')
20:                 valid=0;
21:         if (!valid)
22:             printf("Mh egkyroi xaraktires eisodou!\n");
23:         else if (len>N)
24:             printf("Megali eisodos!\n");
25:         else if ( (x = atol(input)) == 0)
26:             printf("Keni i mideniki eisodos!\n");
27:         else
28:             break;
29:     } while(1);
```

¹ Νάρκισσος ονομάζεται ένας αριθμός που είναι ίσος με το άθροισμα των ψηφίων του υψωμένο το καθένα στο πλήθος των ψηφίων του. Για παράδειγμα, οι αριθμοί: $5 = 5^1$, $153 = 1^3 + 5^3 + 3^3$, $371 = 3^3 + 7^3 + 1^3$, $1634 = 1^4 + 6^4 + 3^4 + 4^4$. Οι Νάρκισσοι αριθμοί μέχρι 9 ψηφία, είναι οι ακόλουθοι: 1, 2, 3, 4, 5, 6, 7, 8, 9, 153, 370, 371, 407, 1634, 8208, 9474, 54748, 92727, 93084, 548834, 1741725, 4210818, 9800817, 9926315, 24678050, 24678051, 88593477, 146511208, 472335975, 534494836, 912985153.



Αριστοτέλειο Πανεπιστήμιο
Θεσσαλονίκης



Τμήμα
Πληροφορικής

SWITCH

Εργαστήριο Λογισμικού και
Διαδραστικών Τεχνολογιών

Χρήστος Κατσάνος
Επίκουρος Καθηγητής
ckatsanos@csd.auth.gr
Γρ. 24, Εθνικής Αντιστάσεως 16,
Καλαμαριά

```
30:
31:   temp = x;
32:
33:   while (temp != 0) {
34:       psifia++;
35:       temp = temp/10;
36:   }
37:
38:   while (temp != 0) {
39:       yp = temp%10;
40:       p = 1;
41:       for (i=1;i<=psifia;i++)
42:           p = p * yp;
43:       sum = sum + p;
44:       temp = temp/10;
45:   }
46:
47:   if (sum == x)
48:       printf("O akeraios %d einai Narkissos.\n", x);
49:   else
50:       printf("O akeraios %d den einai Narkissos.\n", x);
51:
52:   return 0;
53: }
```

Ζητούμενα:

1. Κατασκευάστε τον γράφο ροής του προγράμματος και υπολογίστε την κυκλωματική πολυπλοκότητα με 3 διαφορετικούς τρόπους. **[20%]**
2. Καταγράψτε τα βασικά μονοπάτια και σχεδιάστε τις περιπτώσεις ελέγχου με βάση την τεχνική δοκιμής βασικών μονοπατιών εκτέλεσης. **[20%]**
3. Εντοπίστε συγκεκριμένα λάθη στον κώδικα (bugs) αξιοποιώντας τις περιπτώσεις ελέγχου που σχεδιάσατε. Ξεκινήστε την απάντησή σας καταγράφοντας για κάθε περίπτωση ελέγχου τα αποτελέσματα (δώστε στιγμιότυπο οθόνης / screenshot) σε σύγκριση με τα αναμενόμενα. Προτείνετε διορθώσεις στον κώδικα για τα λάθη που εντοπίσατε. **[10%]**

Σημειώσεις και διευκρινήσεις:

- Προσοχή! Για να είναι ξεκάθαρο ποιες εντολές του κώδικα αντιστοιχούν σε κάθε κόμβο του γράφου ελέγχου θα πρέπει να αναφέρετε μέσα σε κάθε κόμβο του γράφου σε ποιες εντολές αντιστοιχεί. Απάντηση που θα έχει μόνο τον γράφο χωρίς επεξήγηση και αρίθμηση δεν λαμβάνεται υπόψη.
- Προσοχή! Για τον εντοπισμό και τη διόρθωση των λαθών στον κώδικα θα πρέπει να καταγράψετε τα αποτελέσματα των περιπτώσεων ελέγχου που έχετε προτείνει. Απάντηση που δεν έχει τέτοια καταγραφή δεν λαμβάνεται υπόψη.