



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: 2023 – 2024

Εργασία 2: Εγκυροποίηση λογισμικού

Γενικές Οδηγίες

- Οι εργασίες κατατίθενται ηλεκτρονικά μέσω της πλατφόρμας elearning.auth.gr. Καταθέτετε ένα ενιαίο αρχείο σε κειμενογράφο της επιλογής σας. Εργασίες που θα κατατεθούν με οποιονδήποτε άλλο τρόπο (π.χ. email) δεν λαμβάνονται υπόψη.
- Η καταληκτική ημερομηνία υποβολής της εργασίας είναι 21/01/2024. 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»).



Α. Έλεγχος γκρι κουτιού και Εκσφαλμάτωση [100%]

Δίνονται οι λειτουργικές προδιαγραφές ενός λογισμικού:

- Κατασκευάστε ένα πρόγραμμα το οποίο να δέχεται ως είσοδο έναν δεκαεξαδικό αριθμό και να επιστρέφει τον αντίστοιχο δεκαδικό αριθμό.
- Το πρόγραμμα θα δέχεται ως είσοδο έναν θετικό δεκαεξαδικό αριθμό με το πολύ 30 ψηφία. Επίσης, επιτρέπονται μόνο κεφαλαίοι χαρακτήρες. Αν δοθούν από το χρήστη περισσότερα ψηφία, τότε αυτά αγνοούνται και λαμβάνονται υπόψη μόνο τα 30 πρώτα. Δεν χρειάζεται να ενημερωθεί ο χρήστης για αυτό. Θα πρέπει όμως στην εμφάνιση του αποτελέσματος να αναγράφεται ο δεκαεξαδικός αριθμός για τον οποίο έγινε η μετατροπή.
- Θα πρέπει να εμφανίζονται μηνύματα λάθους για τις περιπτώσεις που δοθεί κενή είσοδος ή μη έγκυρο δεκαεξαδικό ψηφίο, και να ζητείται εκ νέου είσοδος από τον χρήστη.
- Το πρόγραμμα θα πρέπει να παράγει σωστό αποτέλεσμα στα όρια του double, αλλιώς να εμφανίζει σχετικό μήνυμα στο χρήστη.

Επίσης, δίνεται παρακάτω μια πρώτη υλοποίηση του λογισμικού σε γλώσσα C. Το αρχείο αυτό είναι διαθέσιμο προς καταβίβαση στο elearning.

```
01: #include <stdio.h>
02: #include <stdlib.h>
03: #include <math.h>
04: #include <string.h>
05: #define N 30
06:
07: int main()
08: {
09:     char input[N];
10:     char *hexarr;
11:     int i,digit,len,validNum;
12:     double num;
13:     printf("Metatropeas apo dekaexadiko se dekadiko systema!\n\n");
14:     do {
15:         printf("Dwste enan dekaexadikos arithmo (mexri %d psifia):",N);
16:         /* Χρησιμοποιούμε την fgets() γιατί σε περίπτωση που ο χρήστης δώσει
           περισσότερους χαρακτήρες από το μέγεθος του πίνακα input δεν θα υπάρξει
           πρόβλημα στο πρόγραμμα */
17:         fgets(input, sizeof input, stdin);
18:
19:         /* Η fgets() όμως δεν σβήνει αυτόματα το \n στο τέλος του string που
           διαβάζει */
20:         input[strlen(input)-1]='\0';
21:         fflush(stdin); // αγνόησε τους υπόλοιπους χαρακτήρες
```



```
22:
23:     len = strlen(input);
24:     validNum = 1;
25:     if (len == 0) {
26:         printf("Den dwsate kapoia psifia.\n\n");
27:         int validNum = 0;
28:     }
29:     else
30:         for (i=0;i<len;i++)
31:             if (input[i]<'0' || input[i]>'F' || (input[i]>'9' && input[i]<'A'))
32:                 {
33:                     printf("To %d psifio den einai egkyro!\n\n",i);
34:                     validNum = 0;
35:                     break;
36:                 }
37: } while (!validNum);
38:
39: hexarr = (char *) malloc(len * sizeof(char));
40:
41: for (i=0;i<len;i++)
42:     hexarr[i] = input[len-i];
43:
44: num = 0.0;
45:
46: for (i=0;i<len;i++) {
47:     if (hexarr[i]<'9')
48:         digit = hexarr[i] - '0';
49:     else
50:         digit = 10 + hexarr[i] - 'A';
51:     num += digit*pow(16,i);
52: }
53: printf("O Dekadikos '%s' antistoixei ston dekadiko '%.0f'.\n\n",input,num);
54: }
```

Ζητούμενα:

Λαμβάνοντας υπόψη μόνο τις λειτουργικές προδιαγραφές του λογισμικού:

1. Σχεδιάστε τις περιπτώσεις ελέγχου που πρέπει να διεξαχθούν με βάση την τεχνική διαμέρισης σε κλάσεις ισοδυναμίας. **[20%]**
2. Σχεδιάστε τις περιπτώσεις ελέγχου που πρέπει να διεξαχθούν με βάση την τεχνική ανάλυσης οριακών τιμών. **[20%]**



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

Λαμβάνοντας υπόψη **μόνο** τον παραπάνω κώδικα του λογισμικού:

3. Κατασκευάστε τον γράφο ροής τους προγράμματος και υπολογίστε την κυκλωματική πολυπλοκότητα με 3 διαφορετικούς τρόπους. **[20%]**
4. Καταγράψτε τα βασικά μονοπάτια και σχεδιάστε τις περιπτώσεις ελέγχου με βάση την τεχνική δοκιμής βασικών μονοπατιών εκτέλεσης. **[20%]**
5. Ομαδοποιήστε τις περιπτώσεις ελέγχου που βρήκατε από την εφαρμογή όλων προηγούμενων τεχνικών. Έπειτα, 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 συγκεκριμένες διορθώσεις στον κώδικα για τα λάθη που εντοπίσατε. **[20%]**

Σημειώσεις και διευκρινήσεις:

- Προσοχή! Θα πρέπει να είναι ξεκάθαρο ποιες εντολές του κώδικα αντιστοιχούν σε κάθε κόμβο του γράφου ελέγχου. 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 με στιγμιότυπα οθόνης (screenshots). Απάντηση που δεν έχει τέτοια καταγραφή δεν λαμβάνεται υπόψη.