

Αξιολόγηση Εργαστηρίου 4ης Εβδομάδος

Εισαγωγή στον Προγραμματισμό (CEID_NY131)

Αξιολόγηση Στόχων
Τα εργαλεία για την
επίτευξη του στόχου

Kleanthis Thramboulidis
Prof. of Software and System Engineering
University of Patras
<https://sites.google.com/site/thramboulidiskleanthis/>



Αξιολόγηση Εργαστηρίου 4ης Εβδομάδος

- Αντικείμενο του Εργαστηρίου (Υλικό)
 - Διαδικασία ανάπτυξης προγράμματος
 - Incremental Development
 - Τα Βήματα E1-E6
 - Αφαιρετικότητα στις διεργασίες
 - Βασικά στοιχεία της Γλώσσας C
- Ήταν ο στόχος εφικτός;
 - Σε πιο βαθμό τον πετύχαμε;
- Σχόλια πάνω στα παραδοτέα

Εργαστήριο 4ης εβδομάδας (Αντικείμενο)

Κατά την διάρκεια του εργαστηρίου της 4^{ης} εβδομάδας θα εκτελέσετε τις παρακάτω Δράσεις:

1. **[Δ1]** Θα αναθέσετε στον υπολογιστή την εύρεση του συχνότερα εμφανιζόμενου (σε συνεχόμενες φορές) αριθμού, σε μία ακολουθία n ακέραιων αριθμών που θα δέχεται από την βασική είσοδο του. (Δες Διαδικασία Ανάπτυξης Προγράμματος s4&5-**MostFrequentNumber**)
2. **[Δ2]** Θα αναθέσετε στον υπολογιστή το έργο της αναγνώρισης πρώτου (prime) αριθμού και την εύρεση των διαιρητών του (divisors) στην περίπτωση που αυτός δεν είναι πρώτος. Τον ακέραιο αριθμό θα δέχεται από την βασική είσοδο και θα ενημερώνει τον χρήστη για το αποτέλεσμα από την βασική έξοδο. (Δες Διαδικασία Ανάπτυξης Προγράμματος s16-**Divisors**)
Περιορισμός : Αξιοποιήστε υποχρεωτικά την αφαιρετικότητα στις διεργασίες.
3. **[Δ3]** Θα εξοικειωθείτε με την λειτουργικότητα του προγράμματος RobotLabyrinth ώστε σε επόμενο εργαστήριο να αναπτύξετε ένα πρόγραμμα (**RemoteRobotController**) το οποίο θα επιτρέπει στο χρήστη να εισάγει ένα ρομπότ μέσα στον λαβύρινθο και να το καθοδηγεί προς την έξοδο.
4. **[Δ4]** Θα βελτιώσετε τον κώδικα των προγραμμάτων που παραδώσατε για το εργαστήριο της 3^{ης} εβδομάδας (Εργαστήριο 3ης εβδομάδας - Αντικείμενο)

👍 18 🗨️ 9

[Δ2] Divisors - Λεκτική περιγραφή 17/10

Αναπτύξτε ένα πρόγραμμα σύμφωνα με το οποίο το σύστημα θα δέχεται ως είσοδο ένα αριθμό και θα ελέγχει αν αυτός είναι πρώτος (prime). Αν ΔΕΝ είναι πρώτος θα εμφανίζει το πλήθος των αριθμών που τον διαιρούν ακριβώς (τους διαιρέτες του).

Λεκτική Περιγραφή – Αφαιρετικότητα στις διεργασίες

- Πάρε τον αριθμό
- Αν ο **αριθμός είναι Πρώτος**
 - Ενημέρωσε τον χρήστη για αυτό
- Αλλιώς
 - **Εμφάνισε διαιρέτες του αριθμού**

Δώστε στην φάση αυτή μόνο την μετατροπή αυτής της ΛΠ σε πηγαίο κώδικα

Υπάρχουν ποικίλες μέθοδοι για να προσδιορίσουμε αν ένας αριθμός n είναι πρώτος. Η πιο βασική μέθοδος, η **δοκιμαστική διαιρέση**, έχει μικρή πρακτική χρησιμότητα επειδή είναι αργή. Ένα τμήμα των σύγχρονων μεθόδων για τον προσδιορισμό αν ένας αριθμός είναι πρώτος είναι εφαρμόσιμο για όλους τους αριθμούς, ενώ οι πιο αποτελεσματικές μέθοδοι είναι διαθέσιμες μόνο για συγκεκριμένες κατηγορίες αριθμών. Οι περισσότερες από αυτές τις μεθόδους λένε μόνο αν ο αριθμός είναι πρώτος ή όχι. Οι μέθοδοι, οι οποίες επιπλέον βρίσκουν και έναν ή περισσότερους παράγοντες του υπό εξέταση αριθμού ονομάζονται αλγόριθμοι παραγοντοποίησης. Πηγή: Wikipedia

Αλγόριθμοι εύρεσης πρώτων – Περιγράφονται 4 αλγόριθμοι

[Δ2] Divisors – ΛΠ -> Πηγαίος Κώδικας

- Πάρε τον αριθμό
- Αν ο **αριθμός είναι Πρώτος**
 - Ενημέρωσε τον χρήστη για αυτό
- Αλλιώς
 - **Εμφάνισε διαιρέτες του αριθμού**

Week 4 – 27/10

<https://eclass.upatras.gr/modules/document/index.php?course=CEID1416&openDir=%2F6512e611b29g>

Function prototypes

- **int isPrime(int num);** //function prototype
- **void displayDivisors(int num);** //function prototype
- **Function declaration?**
 - πρώτος αριθμός (ή απλά πρώτος) είναι ένας φυσικός αριθμός με την ιδιότητα οι μόνοι φυσικοί διαιρέτες του να είναι η μονάδα και ο εαυτός του.

```
if ( isPrime(num) )
    printf( );
else
    displayDivisors(num);
```

[Δ2] Divisors – Program structure

Week3 **17/10** Lecture Time

<https://eclass.upatras.gr/modules/document/file.php/CEID1416/Resources/Divisors.pdf>

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int isPrime(int num);
5 void displayDivisors(int num);
6
7 /* run this program using the console
8 int num;
9
10 int main(int argc, char *argv[]) {
11     printf(...);
12     scanf(...);
13     if(isPrime(num))
14         printf(...);
15     else
16         displayDivisors(num);
17     return 0;
18 }
19
20 int isPrime(int num){
21
22 }
23
24 void displayDivisors(int num){
25
26 }
```

[Δ2] Divisors – Ερωτηματολόγιο

- Η Δράση Δ2 περιλαμβάνει την ανάθεση στη μηχανή του έργου της αναγνώρισης αν ένας αριθμός είναι πρώτος. Υπάρχουν διάφοροι αλγόριθμοι που περιγράφουν τα βήματα της διεργασίας αυτής.
- Από τους παρακάτω [αλγορίθμους που αναφέρει η Wikipedia](#) σημειώστε ποιόν/ποιους από αυτούς έχετε υλοποιήσει ή σκοπεύετε να υλοποιήσετε. Ο καθένας μας οδηγεί σε μια διαφορετική έκδοση του προγράμματος Divisors.

<https://eclass.upatras.gr/modules/questionnaire/pollparticipate.php?course=CEID1416&UseCase=1&pid=1501>

Από: Δευτέρα, 30 Οκτωβρίου 2023 - 8:26 π.μ.

έως: Δευτέρα, 30 Οκτωβρίου 2023 - 11:59 μ.μ.

[Δ2] Divisors – Ερωτηματολόγιο

Ερώτηση Νο 1 – Δράση Νο 2. Σημειώστε ποιόν/ποιους από τους αλγόριθμους για την Δ2 έχετε υλοποιήσει ή σκοπεύετε να υλοποιήσετε.

- Απλός-1
- Απλός-2
- Απλός-3
- Απλός-4

Από: Δευτέρα, 30 Οκτωβρίου 2023 - 8:26 π.μ.

έως: Δευτέρα, 30 Οκτωβρίου 2023 - 11:59 μ.μ.

| Ομάδα | | Απλός-1 | Απλός-2 | Απλός-3 | Απλός-4 |
|---------------|---------------|-----------|-----------|----------|----------|
| No 1 | 29/53 | 18 | 11 | 5 | 2 |
| No 2 | 26/53 | 21 | 9 | 3 | 0 |
| Σύνολο | 55/106 | 39 | 20 | 8 | 2 |

[Δ2] Divisors – Αλγόριθμοι εύρεσης πρώτων

Απλός 1 - από τον ορισμό του πρώτου αριθμού [Επεξεργασία | επεξεργασία κώδικα]

- Εξετάζουμε διαδοχικά όλους τους ακέραιους $M < N$
- Μόλις βρεθεί διαιρέτης του N σταματάμε και ο N δεν είναι πρώτος
- Αν εξαντληθούν οι M χωρίς να βρεθεί διαιρέτης, τότε ο N είναι πρώτος

Απλός 2 [Επεξεργασία | επεξεργασία κώδικα]

Βασιζόμενοι στην παρατήρηση ότι κανένας αριθμός N δεν έχει διαιρέτη μεγαλύτερο του $N/2$, τροποποιούμε τον παραπάνω αλγόριθμο εξετάζοντας όλους τους αριθμούς $M < N/2$, διπλασιάζοντας έτσι την ταχύτητα σε σχέση με τον "Απλό 1".

Απλός 3 [Επεξεργασία | επεξεργασία κώδικα]

Παρατηρούμε ότι αν ένας αριθμός N δεν είναι πρώτος τότε έχει (τουλάχιστον) δύο διαιρέτες μεγαλύτερους από 1 . Σε αυτήν την περίπτωση τουλάχιστον ένας διαιρέτης είναι μικρότερος από την τετραγωνική ρίζα του αριθμού. Τροποποιούμε τον αλγόριθμο 2 εξετάζοντας όλους τους αριθμούς M που είναι μικρότεροι από την τετραγωνική ρίζα του N , αν η τελευταία δεν είναι ακέραιος. Αλλιώς ο αριθμός δεν είναι πρώτος, επειδή τον διαιρεί και η τετραγωνική του ρίζα.

Πηγή: Wikipedia

[Δ2] Divisors – Τελικό Παραδοτέο

Ερώτηση: 2

Δράση Νο 2

Θα δώσετε:

1. Την λεκτική περιγραφή (σε μορφή κειμένου) μιας από τις εκδόσεις του προγράμματος divisors την οποία ολοκληρώσατε. Αναφορά στις εκδόσεις έγινε στο αντίστοιχο Ερωτηματολόγιο.
2. Τον πηγαίο κώδικα των εκδόσεων που υλοποιήσατε. Θα δώσετε τον σύνδεσμο σε ένα αρχείο πηγαίου κώδικα (αν έχετε όλες τις εκδόσεις σε ένα αρχείο με σχόλια) ή ένα σύνδεσμο για κάθε αρχείο πηγαίου κώδικα που αντιστοιχεί σε έκδοση του προγράμματος που υλοποιήσατε.

[Δ1] – MostFrequentNumber (MFN)

17/10

Αναθέστε στον υπολογιστή το παρακάτω έργο:

Να δέχεται από την κύρια είσοδο (standard input - stdin) μία ακολουθία από n αριθμούς και θα εμφανίζει στην κύρια έξοδο (standard output - stdout):

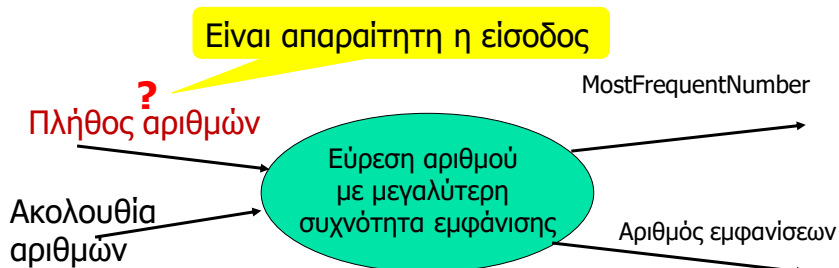
- τον αριθμό που εμφανίστηκε στην ακολουθία περισσότερες συνεχόμενες φορές, και
- το πλήθος των εμφανίσεων του.

[Δ1] - MFN – Ενέργειες E1, E2

17/10

Συχνότερα εμφανιζόμενος αριθμός

- Αναπτύξτε ένα πρόγραμμα σύμφωνα με το οποίο το σύστημα θα διαβάζει μία ακολουθία από n αριθμούς και θα εμφανίζει: α) τον αριθμό που εμφανίστηκε στην ακολουθία περισσότερες συνεχόμενες φορές και β) το πλήθος των συνεχόμενων εμφανίσεων του.



[Δ1] - MFN – Ε3(Λεκτική Περιγραφή) 2/3 ^{17/10}

62 αριθμός 12 12 91 62 62 62 120 120 57 ...

62 Υποψήφιος Συχνότερος Αριθμός 3 Πλήθος Εμφανίσεων ΥΣΑ

12 Συχνότερος Αριθμός 2 Πλήθος Εμφανίσεων ΣΑ

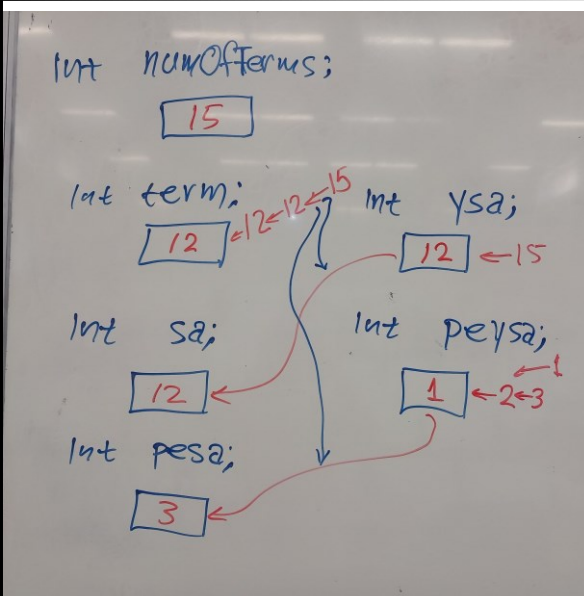
....
Για κάθε επόμενο αριθμό της ακολουθίας

- **Αν** αυτός είναι ίσος με τον ΥΣΑ
 - Αύξησε κατά 1 το ΠΕΥΣΑ
 - **Αν** ο ΠΕΥΣΑ είναι μεγαλύτερος από το ΠΕΣΑ
 - Αντικατέστησε τον ΣΑ με τον ΥΣΑ
 - Αντικατέστησε το ΠΕΣΑ με ΠΕΥΣΑ
- **Αλλιώς**
 - Αντικατέστησε τον ΥΣΑ με τον αριθμό

Βρείτε λάθη
Βελτιώστε τον

Εμφάνιση ΣΑ και ΠΕΣΑ


[Δ1] - MFN – Η βάση για την ΛΠ



Week 4 – 27/10

<https://eclass.upatras.gr/modules/document/file.php/CEID1416/Resources/Week4-MostFrequentNumber1.jpg>

[Δ1] MFN – Τελικό Παραδοτέο

Ερώτηση: 1 

Δράση Νο 1

Θα δώσετε:

1. Την λεκτική περιγραφή (σε μορφή κειμένου) μιας από τις εκδόσεις του προγράμματος MostFrequentNumber την οποία ολοκληρώσατε.
2. Τον σύνδεσμο σε ένα αρχείο πηγαίου κώδικα που αντιστοιχεί στην παραπάνω λεκτική περιγραφή.
3. Ένα screenshot από την εκτέλεση του προγράμματος.

Τελικό Παραδοτέο

- Ομάδα 1 - 44/53 υποβολές
- Ομάδα 2 - 39/53 υποβολές

