



Εισαγωγή στον Προγραμματισμό

Εργασία #0

Οκτώβριος 2025

Στόχος της εργασίας είναι να αποκτήσουμε εμπειρία με τα βασικά εργαλεία του προγραμματιστή και να γράφουμε ένα από τα πρώτα προγράμματα σε C. Συγκεκριμένα, στοχεύουμε γνωριμία με τα ακόλουθα:

1. git, version control, repositories, GitHub.
2. Γραμμή εντολών Linux και bash.
3. Χρήση ακεραίων στην C.
4. Χρήση βασικών δομών (loops, if, functions).
5. Δεδομένα εισόδου από πρότυπη είσοδο.

Υποβολή Εργασίας. Όλες οι υποβολές των εργασιών θα γίνουν μέσω GitHub και συγκεκριμένα στο github.com/progintro [2]. Προκειμένου να ξεκινήσεις, μπορείς να δεχτείς την άσκηση με αυτήν την: πρόσκληση [3].

1. Νέο URL στο GitHub (pages - 30 Μονάδες)

Έχεις λογαριασμό στο GitHub; Έχεις ξαναφτιάξει repository; Έχεις φτιάξει URL στο διαδίκτυο; Μετά από αυτήν την άσκηση θα μπορείς να απαντήσεις "ναι" σε όλα. Αν ήδη μπορείς να απαντήσεις ναι, η άσκηση αυτή (ελπίζουμε) να μην σου πάρει πολύ χρόνο.

Η ιστοσελίδα του μαθήματος βασίζεται σε ένα public GitHub repository [1]. Σκοπός αυτής της άσκησης είναι να δημιουργήσεις ένα καινούριο repository στον προσωπικό σου λογαριασμό GitHub και να το συνδέσεις με GitHub Pages προκειμένου να είναι διαθέσιμο στο διαδίκτυο. Αυτό το tutorial [4] μπορεί να σε βοηθήσει.

Τεχνικές Προδιαγραφές

Η υποβολή σου, πρέπει να έχει τα ακόλουθα χαρακτηριστικά:

- Repository Name: progintro/hw0-<YourUsername>
- Αρχείο Λύσης Filepath: pages/solution.txt
- README Filepath (optional): pages/README.md

Έστω ότι το GitHub username σου είναι YourUsername. Το αρχείο solution.txt πρέπει να περιέχει το URL με το domain YourUsername.github.io που δημιουργήσατε. Αν χρησιμοποιήσετε κάποιο *.github.io URL που δεν δημιουργήσατε η άσκηση μηδενίζεται. Ενδεικτική συμπεριφορά για μια επιτυχημένη υποβολή (**προσοχή: η γραμμή με την εντολή curl εκτείνεται σε δύο γραμμές, πρέπει να αντιγράψετε και τις δύο προκειμένου να τρέξετε αυτήν την εντολή¹**):

```
# Check out solution URL
$ cat solution.txt
YourUsername.github.io
# Ensure the URL exists
$ curl --output /dev/null --silent --head --fail YourUsername.github.io && \
  echo "URL exists" || echo "URL does not exist"
URL exists
```

Το αρχείο README.md είναι προαιρετικό αν θέλετε να προσθέσετε κάτι στην υποβολή σας.

2. Παιχνίδια με Κονσόλα (cmdline - 150 Μονάδες)

Έχεις χρησιμοποιήσει την γραμμή εντολών; Linux; Γνωρίζεις πως να κάνεις SSH σε απομακρυσμένους servers και να λύνεις προβλήματα με εντολές κονσόλας; Αν όχι, καιρός να μάθουμε! Αν ναι, τότε αυτή η άσκηση θα σου είναι παιχνιδάκι.

Σε αυτήν την άσκηση θα μπούμε με ssh σε έναν απομακρυσμένο server που βρίσκεται στην IP διεύθυνση 35.169.96.19 μέσω τερματικού και θα ανακτήσουμε συγκεκριμένες πληροφορίες. Υπάρχουν διαφορετικοί χρήστες (byte0, byte1, ...) που αντιστοιχούν σε διαφορετικά προβλήματα/υποερωτήματα. Για όλους τους χρήστες ο κωδικός για να μπειτε είναι bits&bytes. Σε κάθε ένα υποερώτημα θα αναζητούμε ένα string που μπορεί να είναι μια μόνο λέξη ή πολλές λέξεις διαχωρισμένες με τον χαρακτήρα underscore _ - για παράδειγμα: something_like_that.

¹Προσοχή στο backslash \ στην γραμμή εντολών, αυτό που θέλουμε να δείξουμε είναι ότι η εντολή είναι μία αλλά εκτείνεται σε δύο γραμμές.

byte0: Αλλάζοντας Φάκελο (cd)

Ο χρήστης byte0 είναι ελαφρώς μανιακός με την οργανωτικότητα. Του αρέσει να οργανώνει όλα του τα αρχεία σε φακέλους, υποφακέλους κ.ο.κ.. Με αυτά και με αυτά όμως, δεν μπορεί να βρει το αρχείο του, μπορείτε να τον βοηθήσετε; Για να μπειτε στον server τρέξτε:

```
ssh byte0@35.169.96.19
```

και βάλτε τον κωδικό που γράψαμε παραπάνω. Η εντολή cd μπορεί να σας φανεί χρήσιμη όπως και η εντολή cat (ίσως και το tab / autocomplete :)). Όταν διαβάσετε τα περιεχόμενα του αρχείου, σημειώστε το string που βρήκατε μέσα στο αρχείο ώστε να το προσθέσετε στο solution.txt (δες παρακάτω στις τεχνικές προδιαγραφές).

byte1: Μια Εντολή Άγνωστη

Ο χρήστης byte1 έφτιαξε ένα καινούριο πρόγραμμα με άγνωστη σε μας λειτουργία ονόματι supercalifragilisticexpialidocious. Πως μπορείτε να βρείτε τι κάνει;

byte2: Τα Άπαντα του Shakespeare (grep)

Ο χρήστης byte2 έχει δύο βασικά χαρακτηριστικά: (1) είναι μεγάλος θαυμαστής του Shakespeare και (2) είναι κάπως μυστικοπαθής. Αποφάσισε να συνδυάσει τα δύο και καταχώνιασε ένα μυστικό string μέσα στην συλλογή του με τα άπαντα του Shakespeare. Μπορείτε να τον βοηθήσετε να το βρεί;

Το μόνο που γνωρίζουμε είναι πως το κρυφό κείμενο περιέχει το string "will find". Η εντολή grep μπορεί να σας φανεί χρήσιμη.

byte3: Τα Άπαντα του Shakespeare Άλλαξαν (diff)

Ο χρήστης byte3 αποφάσισε να αλλάξει κάτι στα άπαντα του Shakespeare ελπίζοντας πως η αλλαγή θα περάσει απαρατήρητη. Ελέγξαμε τον αριθμό των λέξεων και χαρακτήρων του κάθε αρχείου:

```
byte3@ip-172-31-37-131:~$ wc -w *.txt
901325 shakespeare.modified.txt
901325 shakespeare.txt
1802650 total
byte3@ip-172-31-37-131:~$ wc -c *.txt
5458203 shakespeare.modified.txt
5458199 shakespeare.txt
```

και παρατηρήσαμε πως ενώ ο αριθμός λέξεων παρέμεινε ίδιος (901.325), ο αριθμός των χαρακτήρων άλλαξε. Επομένως, υποψιαζόμαστε πως άλλαξε μια λέξη του αυθεντικού με μια καινούρια. Βρείτε την καινούρια λέξη—η εντολή diff ίσως σας φανεί χρήσιμη.

byte4: Λαβύρινθος (find)

Ο χρήστης byte4 είναι συγγενής του byte2 παραπάνω και ελαφρώς πιο μυστικοπαθής. Προκειμένου να "ανεβάσει" το παιχνίδι του αποφάσισε να κρύψει τα δεδομένα του σε ένα δαιδαλώδες κατασκεύασμα από φακέλους. Μπορείτε και πάλι να τον βοηθήσετε να βρει το μυστικό που έχει κρύψει μέσα σε ένα από όλα τα αρχεία των υποφακέλων; Το μόνο που γνωρίζουμε είναι πως έχει κρύψει το μυστικό του σε ένα αρχείο που λέγεται cup.txt. Η εντολή find μπορεί να σας φανεί χρήσιμη.

byte5: Compile and Run (gcc, mkdir, cp)

Ο byte5 έγραψε ένα προγραμματάκι σε C ονόματι byte5.c. Μεταγλωττίστε το και τρέξτε το προκειμένου να πάρετε τον κωδικό σας! Προσοχή: ο default φάκελος του byte5 δεν σας επιτρέπει να χρησιμοποιήσετε αρχεία, ίσως χρειαστεί να φτιάξετε έναν καινούριο υποφάκελο μέσα στον φάκελο /tmp προκειμένου να τοποθετήσετε το εκτελέσιμό σας. Οι εντολές mkdir, cp, gcc μπορεί να σας φανούν χρήσιμες.

byte6: Αποσυμπίεση αρχείου 1 (unzip)

Κάποιες φορές προκειμένου να μικρύνουμε τον χώρο που πιάνει ένα αρχείο το συμπιέζουμε. Ο χρήστης byte6 έχει ένα αρχείο byte6.zip το οποίο χρειάζεται αποσυμπίεση. Ανοίξτε το για να αποκτήσετε πρόσβαση στα περιεχόμενά του. Το πρόγραμμα unzip μπορεί να σας βοηθήσει.

byte7: Αποσυμπίεση αρχείου 2 (tar)

Υπάρχουν διάφοροι τρόποι και εργαλεία για να συμπίεσεις αρχεία και τα περιεχόμενά τους. Ο χρήστης byte7 χρησιμοποίησε ένα πρόγραμμα που λέγεται tar - μπορείτε να χρησιμοποιήσετε το ίδιο προκειμένου να βρείτε τα περιεχόμενα του byte7.tar.gz;

byte8: Carriage Return (xxd, pico, vim)

Ο χρήστης byte8 μας είπε πως ο κωδικός που ψάχνουμε είναι μέσα στο αρχείο carriage_return.txt. Όμως δοκιμάσαμε να το τυπώσουμε:

```
byte8@ip-172-31-37-131:~$ ls
carriage_return.txt
byte8@ip-172-31-37-131:~$ cat carriage_return.txt
There is absolutely nothing to see here. Move along.
```

και δεν είδαμε κάτι. Μπορείς να μας βοηθήσεις; Πιστεύουμε πως οι εντολές xxd, pico ή vim μπορεί να φανούν χρήσιμες.

byte9: Ένα Περιεργό Όνομα (cat)

Ο χρήστης byte9 ανέφερε πως ο κωδικός του είναι μέσα στον φάκελο /home/byte9 - μπορείτε να μας βοηθήσετε να τον ανακτήσουμε; Ίσως σε βοηθήσει να θυμηθείς τι είναι το filepath (μονοπάτι) ενός αρχείου.

byte10: Το 42ο Όνομα (sort, head)

Ο χρήστης byte10 έχει ένα αρχείο γεμάτο με ονόματα και θέλει να τα ταξινομήσει αλφαβητικά και να βρει το 42ο όνομα στην σειρά. Γνωρίζει ότι το 1ο όνομα είναι το Aarika - μπορείτε να τον βοηθήσετε να βρει το 42ο; Οι εντολές sort και head μπορούν να συνδυαστούν και να βοηθήσουν.

byte11: Επιλογές Ονόματος (sort, uniq)

Ο χρήστης byte11 μόλις απέκτησε ένα παιδάκι και προκειμένου να μην το κοροϊδεύουν στο σχολείο αποφάσισε να του δώσει το πιο συχνό όνομα εκείνης της χρονιάς. Στο αρχείο births.txt περιέχονται τα ονόματα όλων των φετινών γεννήσεων. Μπορείτε να τον βοηθήσετε να βρει το πιο συχνό όνομα; Οι εντολές sort και uniq ίσως σας φανούν χρήσιμες.

byte12: Βρες το Μονοπάτι (which, PATH)

Ο byte12 είναι κάπως αφηρημένος. Μόλις έφτιαξε το πρόγραμμα tuxsay αλλά δεν θυμάται που το τοποθέτησε και δεν πρόλαβε να φτιάξει ένα manual page. Παρόλα αυτά, μπορεί να το τρέξει κανονικά. Μπορείτε να βρείτε το basename του φακέλου μέσα στον οποίο βρίσκεται το πρόγραμμα tuxsay;

byte13: Commit Messages (git log)

Ο byte13 μόλις πήρε πρόσβαση σε ένα git repository (αποθετήριο) και θέλει να δει τα commit messages (μηνύματα για προηγούμενες αλλαγές) που έκαναν οι χρήστες που συνεισέφεραν στο repository. Μπορείτε να διαβάσετε αυτά τα μηνύματα;

byte14: Ιστορικό (git log -p)

Ο byte14 ξεκίνησε να γράφει μια νουβέλα με την βοήθεια του ΑΙ. Μετά από αρκετές αλλαγές διαπίστωσε ότι το ΑΙ είχε προσθέσει (κατά λάθος;) έναν από τους κωδικούς του μέσα στο κείμενο. Ευτυχώς το είδε, το αφαίρεσε, έκανε commit τις αλλαγές και πλέον ξέρει δεν είναι ορατό σε άλλους χρήστες. Έχει δίκιο ο byte14;

byte15: Open the Vault (bonus / προαιρετικό)

Ο byte15 είναι χρήστης του vault, ενός υπερασφαλούς προγράμματος που εμφανίζει τα περιεχόμενα του λογαριασμού σου, μόνο εφόσον γνωρίζεις το μυστικό PIN. Δυστυχώς, ο byte15 ξέχασε το PIN του! Μπορείτε να τον βοηθήσετε να ανακτήσει τα περιεχόμενα του λογαριασμού του;

Τεχνικές Προδιαγραφές

Η υποβολή σου, πρέπει να έχει τα ακόλουθα χαρακτηριστικά:

- Repository Name: progintro/hw0-<YourUsername>
- README Filepath: cmdline/README.md
- Αρχείο Λύσης Filepath: cmdline/solution.txt
- **Κάθε γραμμή που θα προστεθεί στο αρχείο solution.txt πρέπει να γίνει με διαφορετικό git commit. Κοινώς, θέλουμε να δούμε τουλάχιστον *15* διαφορετικά git commits. Η κάθε γραμμή πρέπει να προστεθεί όταν λύσετε το αντίστοιχο πρόβλημα, όχι όλες μαζί στο τέλος.**

Το περιεχόμενο του solution.txt πρέπει να έχει την ακόλουθη μορφή:

```
$ cat solution.txt
byte0: ...
byte1: ...
byte2: ...
...
```

Όπου οι τελείες (...) περιγράφουν το μέρος της λύσης που έχει παραληφθεί.

Στο αρχείο README.md πρέπει να γράψετε μια σύντομη περιγραφή για τον τρόπο που λύσατε το κάθε πρόβλημα.

Αν σου άρεσε αυτή η άσκηση ή έχεις κολλήσει με ένα από τα παραπάνω και δεν ξέρεις πως να συνεχίσεις, υπάρχουν και άλλα site online στα οποία μπορείς να εξασκηθείς όπως για παράδειγμα το overthewire [5] τα οποία μπορούν να σου δώσουν ιδέες. Σημείωση: μέχρι το επίπεδο 11 τα προβλήματα σε αυτό το site είναι εντός θέματος για το μάθημα, από εκεί και πέρα γίνονται πιο δύσκολα / απαιτούν γνώσεις που θα αποκτήσουμε σε μεγαλύτερα έτη.

3. Οι Ακολουθίες Aliquot (180 Μονάδες)

Στα μαθηματικά, η ακολουθία aliquot [7] είναι μια ακολουθία θετικών ακεραίων στην οποία ο κάθε όρος είναι το άθροισμα των γνήσιων διαιρετών² του προηγούμενου όρου. Αν η ακολουθία φτάσει στον αριθμό 1, τερματίζει, εφόσον το άθροισμα των γνήσιων διαιρετών του 1, είναι 0. Η ακολουθία μπορεί να ξεκινήσει με οποιονδήποτε θετικό ακέραιο n . Ας υπολογίσουμε την ακολουθία aliquot για τον αριθμό $n = 12$. Συμβολίζοντας με $s(n)$ το άθροισμα των γνήσιων διαιρετών του n έχουμε:

$$s(12) = 1 + 2 + 3 + 4 + 6 = 16$$

καθώς οι 1, 2, 3, 4, 6 είναι όλοι οι γνήσιοι διαιρέτες του 12. Επίσης, εφαρμόζοντας τον ίδιο υπολογισμό στο 16 (τον 2ο όρο της ακολουθίας παίρνουμε:

$$s(s(12)) = s(16) = 1 + 2 + 4 + 8 = 15$$

Συνεχίζοντας με τον ίδιο τρόπο:

$$s(s(s(12))) = s(15) = 1 + 3 + 5 = 9$$

$$s(s(s(s(12)))) = s(9) = 1 + 3 = 4$$

$$s(s(s(s(s(12))))) = s(4) = 1 + 2 = 3$$

$$s(s(s(s(s(s(12)))))) = s(3) = 1$$

$$s(s(s(s(s(s(s(12))))))) = s(1) = 0$$

Παρατηρούμε ότι η ακολουθία τελικά φτάνει στο 0 μετά από 7 βήματα ($12 \rightarrow 16 \rightarrow 15 \rightarrow 9 \rightarrow 4 \rightarrow 3 \rightarrow 1 \rightarrow 0$). Ο αριθμός των βημάτων της ακολουθίας aliquot (ουσιαστικά ο αριθμός των μεταβάσεων \rightarrow) που απαιτούνται για να μεταβεί ένας ακέραιος n στο 0 λέγεται και *μήκος aliquot* για το δοθέν n (aliquot length for n).

Απλοποιούμε τις εμφωλευμένες παρενθέσεις παραπάνω, με τον συμβολισμό:

$$s^1(n) = s(n),$$

$$s^2(n) = s(s(n)),$$

$$s^3(n) = s(s(s(n))),$$

$$\vdots$$

$$s^m(n) = \underbrace{s(s(\cdots s(n) \cdots))}_{m \text{ φορές}} .$$

που μας επιτρέπει να γράψουμε $s^2(12) = 15$ ή $s^7(12) = 0$. Ισοδύναμα, λέμε ότι το μήκος aliquot l είναι ο μικρότερος ακέραιος l για τον οποίο $s^l(n) = 0$.

²Γνήσιοι διαιρέτες (proper divisors [8]) ενός αριθμού n λέγονται όλοι οι ακέραιοι διαιρέτες του n πλην του εαυτού του.

Στο παραπάνω παράδειγμα, παρατηρήσαμε ότι για το 12 η ακολουθία aliquot ήταν φθίνουσα, δηλαδή εφαρμόζοντας το s επανειλημμένα οδηγούμαστε προς μικρότερους αριθμούς και τελικά στο 0. Είναι όμως όλες οι ακολουθίες aliquot φθίνουσες; Η απάντηση είναι όχι, καθώς υπάρχουν αριθμοί όπως το 6, όπου:

$$s(6) = 1 + 2 + 3 = 6$$

Τέτοιοι αριθμοί λέγονται τέλει (perfect numbers [9]) και οι ακολουθίες aliquot δεν τερματίζουν ποτέ για αυτούς καθώς $s^m(6) = 6$ για κάθε m . Τέτοια παραδείγματα κύκλων στις ακολουθίες aliquot δεν περιορίζονται μόνο στους τέλει αριθμούς, για παράδειγμα υπάρχουν και οι φίλιοι αριθμοί (amicable numbers [6]) όπως είναι οι 220 και 284. Συγκεκριμένα:

$$\begin{aligned} s^2(220) &= s(s(220)) = s(1 + 2 + 4 + 5 + 10 + 11 + 20 + 22 + 44 + 55 + 110) \\ &= s(284) = 1 + 2 + 4 + 71 + 142 = 220 \end{aligned}$$

Η ακολουθία aliquot για τον ακέραιο 220 είναι: $220 \rightarrow 284 \rightarrow 220 \rightarrow \dots$ και εναλλάσσεται ανάμεσα σε αυτούς τους δύο αριθμούς χωρίς να τερματίζει ποτέ. Αντίστοιχα, υπάρχουν και άλλες κατηγορίες αριθμών όπως είναι οι κοινωνικοί αριθμοί (sociable numbers [10]) οι οποίοι έχουν κύκλους μεγαλύτερης περιοδικότητας.

Επομένως, μια ακολουθία aliquot φαίνεται να "τερματίζει" με έναν από τους ακόλουθους δύο τρόπους: (1) είτε θα φτάσει στο 0, (2) είτε θα πέσει σε κάποιον κύκλο με τέλειους / φίλιους / κοινωνικούς αριθμούς. Αυτή είναι και η υπόθεση των Catalan-Dickson [7], μια εικασία που κρατάει από το 1888 και ακόμα δεν έχουμε καταφέρει να αποδείξουμε αν ισχύει ή όχι, παρόλες τις τεχνολογικές και θεωρητικές εξελίξεις³. Μάλιστα για κάποιους—σχετικά μικρούς—αριθμούς όπως είναι ο 276 [11] δεν έχουμε ακόμα καταφέρει να υπολογίσουμε την πλήρη ακολουθία!

Για το ζητούμενο αυτής της άσκησης θα χρειαστεί να υλοποιήσετε ένα πρόγραμμα το οποίο να μπορεί να υπολογίζει τους όρους και τα μήκη της ακολουθίας aliquot για τους αριθμούς που επιλέγει ο χρήστης. Φυσικά, θα υπάρχουν περιορισμοί στο τι μπορεί να περιμένει ο χρήστης από το πρόγραμμά σας, οι οποίοι καταγράφονται στις τεχνικές προδιαγραφές που ακολουθούν.

Τεχνικές Προδιαγραφές

- Repository Name: progintro/hw0-<YourUsername>
- Αρχείο C (Filepath): aliquot/src/aliquot.c
- Το αρχείο C που θα υποβληθεί πρέπει να μεταγλωττίζεται χωρίς ειδοποιήσεις για λάθη και με κωδικό επιστροφής (exit code) που να είναι 0. Συγκεκριμένα,

³Αν τυχόν καταφέρετε να αποδείξετε την εικασία στα πλαίσια αυτής της άσκησης, παρακαλώ επικοινωνήστε με τους διδάσκοντες.

το αρχείο σας **πρέπει** να μπορεί να μεταγλωττιστεί επιτυχώς με την ακόλουθη εντολή σε ένα από τα μηχανήματα του εργαστηρίου (linuxXY.di.uoa.gr):

```
gcc -O0 -m32 -Wall -Wextra -Werror -pedantic -o aliquot aliquot.c
```

- README Filepath: aliquot/README.md
- Το πρόγραμμά σας πρέπει να διαβάσει από την πρότυπη είσοδο (stdin) 3 διαφορετικές τιμές:
 1. Τον θετικό ακέραιο από τον οποίο θέλετε να ξεκινήσετε την ακολουθία.
 2. Το μέγιστο μήκος της ακολουθίας (το 0 θα συμβολίζει ότι επιτρέπεται ακολουθίες "άπειρου" μήκους) που θέλουμε να υπολογιστεί.
 3. Έναν χαρακτήρα που να δηλώνει αν θέλετε το πρόγραμμά σας να τυπώσει την πλήρη ακολουθία 'f' (full) ή να τυπώσει μόνο το μήκος της 'l' (length).
- Δείτε παραδείγματα από τις πρότυπες εκτελέσεις παρακάτω για να δείτε την αναμενόμενη έξοδο.
- Όλοι οι ακέραιοι που θα δοθούν στο πρόγραμμά σας θα είναι ακέραιοι μικρότεροι από 10^{15} . Αν κατά την διαδικασία του υπολογισμού της ακολουθίας το πρόγραμμά σας φτάσει σε ακέραιο μεγαλύτερο του 10^{15} , το πρόγραμμά σας πρέπει να τερματίσει με μήνυμα λάθους και κωδικό εξόδου (exit code) 1.
- Αν δοθεί οποιαδήποτε είσοδος εκτός προδιαγραφών (για παράδειγμα ο χρήστης δεν δώσει κάποιον ακέραιο για να υπολογιστεί η ακολουθία) το πρόγραμμά σας πρέπει να τερματίζει με κωδικό εξόδου (exit code) 1.
- Για μια ακολουθία μήκους n , οι χρήστες αναμένουν πως το πρόγραμμά σας πρέπει να ολοκληρώνει την εκτέλεσή του μέσα σε λιγότερο από n δευτερόλεπτα.

Παρακάτω παραθέτουμε αλληλεπιδράσεις με μια ενδεικτική λύση:

```
$ hostname
linux14
$ gcc -O0 -m32 -Wall -Wextra -Werror -pedantic -o aliquot aliquot.c
$ ./aliquot
Please give the number to start the aliquot sequence from: 12
Provide the max aliquot length to look for (0 for unlimited): 0
Do you want to print the full sequence ('f') or just the length ('l')? f
12
16
15
9
```

```

4
3
1
0
$ ./aliquot
Please give the number to start the aliquot sequence from: 12
Provide the max aliquot length to look for (0 for unlimited): 0
Do you want to print the full sequence ('f') or just the length ('l')? l
Length of aliquot sequence: 7
$ echo $?
0
$ ./aliquot
Please give the number to start the aliquot sequence from: 138
Provide the max aliquot length to look for (0 for unlimited): 200
Do you want to print the full sequence ('f') or just the length ('l')? l
Length of aliquot sequence: 178
$ ./aliquot
Please give the number to start the aliquot sequence from: 6
Provide the max aliquot length to look for (0 for unlimited): 6
Do you want to print the full sequence ('f') or just the length ('l')? f
6
6
6
6
6
6
6
$ ./aliquot
Please give the number to start the aliquot sequence from: 276
Provide the max aliquot length to look for (0 for unlimited): 0
Do you want to print the full sequence ('f') or just the length ('l')? f
276
396
...
749365894850244
1414070378301756
Number exceeds maximum supported integer (10000000000000000). Stopping.
$ echo $?
1

```

Στο αρχείο README.md πρέπει να προσθέσετε οποιεσδήποτε παρατηρήσεις σας κατά την διεκπεραίωση της άσκησης. Ο κώδικας απαιτείται να είναι καλά τεκμηριωμένος με σχόλια καθώς αυτό θα είναι μέρος της βαθμολόγησης.

Αναφορές

- [1] DIT. Repository για το μάθημα . <https://github.com/progintro/progintro.github.io>.
- [2] DIT. Οργανισμός για το μάθημα (GitHub progintro) . <https://github.com/progintro>.
- [3] DIT. Πρόσκληση για Εργασία 0 . <https://classroom.github.com/a/XwPvT9Jk>.
- [4] Github. GitHub Pages tutorial . <https://docs.github.com/en/pages/quickstart>.
- [5] OverTheWire. Ασκήσεις σε Γραμμή Εντολών . <https://overthewire.org/wargames/bandit/bandit0.html>.
- [6] Wikipedia. Amicable Numbers . https://en.wikipedia.org/wiki/Amicable_numbers.
- [7] Wikipedia. Aliquot Sequence. https://en.wikipedia.org/wiki/Aliquot_sequence.
- [8] Wikipedia. Divisor. <https://en.wikipedia.org/wiki/Divisor>.
- [9] Wikipedia. Perfect Number. https://en.wikipedia.org/wiki/Perfect_number.
- [10] Wikipedia. Sociable Number. https://en.wikipedia.org/wiki/Sociable_number.
- [11] YouTube. The Amazing Number 276 (αν ξεκινήσατε την άσκηση μια μέρα (ή ώρα) πριν την υποβολή συνιστούμε να μην πατήσετε το λινκ). <https://www.youtube.com/watch?v=OtYKDzXwDEE>.