

Διαδικασία Ανάπτυξης Προγράμματος

Εισαγωγή στον Προγραμματισμό (CEID_NY131)

Διαδικασία Ανάπτυξης
Προγράμματος
(Program Development
Process)

με τη χρήση παραδειγμάτων

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



Οργάνωση Διάλεξης

- Τα παραδείγματα
 - MostFrequentNumber
 - AddNumbers
- Η οργάνωση του αποθηκευτικού χώρου
- Διαδικασία Ανάπτυξης Προγράμματος
 - Τα βήματα
 - Η Λεκτική Περιγραφή
 - Απλά Παραδείγματα (Hailstone, Fibonacci, Divisors)
- MostFrequentNumber - Διαδικασία Ανάπτυξης
 - Διαδοχικές εκδόσεις βελτίωσης του αλγορίθμου
- AddNumbers - Διαδικασία Ανάπτυξης
 - Incremental Development
- Βασικά στοιχεία της γλώσσας C

Τα παραδείγματα μας 1/2

- HelloWorld
 - [I2P_BasicConceptsPartA](#) (s25-28)
- Add2Numbers
 - [I2P_BasicConceptsPartA](#) (s13), [I2P_BasicConceptsPartC](#) (s4-13)
- HailstoneSequence
 - [I2P_BasicConceptsPartB](#) (s11-15)
- FibonacciNumbers
 - [I2P_BasicConceptsPartB](#) (s23-26)

Τα παραδείγματα μας 2/2

- Average
 - Ως πρόγραμμα [I2P_BasicConceptsPartC](#) (s17-23)
 - Ως συνάρτηση (function)
[I2P_BasicConceptsPartC](#) (s24-27)
- MostFrequentNumber
 - I2P_ProgramDevelopmentUsingExamples (Current) (s5)
- AddNumbers
 - [I2P_BasicConceptsPartC](#) (s28)
 - I2P_ProgramDevelopmentUsingExamples (Current)
- Divisors
 - I2P_ProgramDevelopmentUsingExamples (Current) (s16)

MostFrequentNumber

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):

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

AddNumbers – Η άσκηση

Η διεργασία πρόσθεσης αριθμών

Γράψτε ένα πρόγραμμα σύμφωνα με το οποίο η μηχανή θα υπολογίζει το άθροισμα ακέραιων αριθμών.

Προσέξτε τους περιορισμούς!

Οι Περιορισμοί:

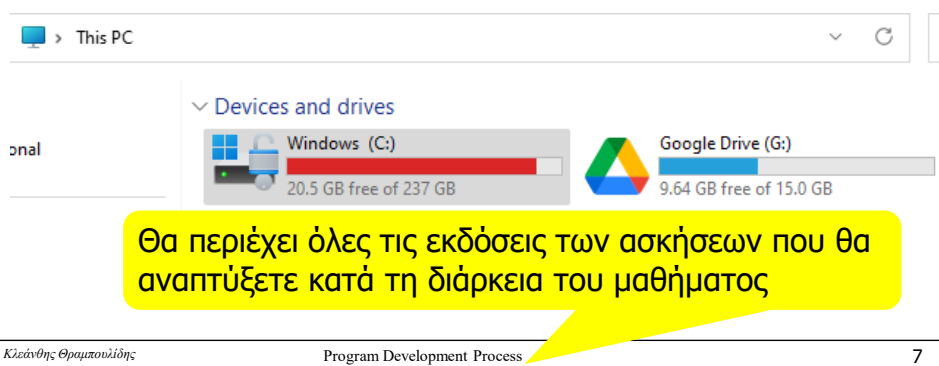
Υποθέστε πως η μηχανή :

- a) δεν μπορεί να υπολογίσει άθροισμα αριθμών με περισσότερα από ένα ψηφία. Για τον υπολογισμό του αθροίσματος 2 ψηφίων χρησιμοποιήστε τον τελεστή +.
- b) μπορεί να κάνει τις πράξεις του πολλαπλασιασμού και διαίρεσης ακέραιων αριθμών με χρήση των τελεστών * και / αντίστοιχα. Επιπλέον μπορεί να βρίσκει το modulo ενός ακεραίου αριθμού με χρήση του τελεστή %.

Η οργάνωση του αποθηκευτικού χώρου 1/3

Οδηγίες για την οργάνωση του αποθηκευτικού χώρου των ασκήσεων εργαστηρίου.

- **Ενιαίος αποθηκευτικός χώρος στο νέφος (Cloud)**
 - Ενδεικτικά μπορείτε να χρησιμοποιήσετε το Google drive ([Οδηγίες αξιοποίησης από το IDE](#))



Θα περιέχει όλες τις εκδόσεις των ασκήσεων που θα αναπτύξετε κατά τη διάρκεια του μαθήματος

Η οργάνωση του αποθηκευτικού χώρου 2/3

Ενιαίος αποθηκευτικός χώρος στο νέφος (Cloud)

- Ενδεικτικό όνομα Ευρετηρίου: **I2P-LabExersices**
- Κάθε άσκηση θα έχει το δικό της ευρετήριο, π.χ. Ευρετήριο HelloWorld, Ευρετήριο Add2Numbers

<< Google Drive (G:) > My Drive > University > Ceid > I2P > I2P-LabExersices

Name	Date modified	Type
Add2NumbersV1	13/10/2023 11:35	File folder
FibonacciNumbers	13/10/2023 11:50	File folder
HailStoneSequence	13/10/2023 11:38	File folder
HelloWorld	13/10/2023 11:53	File folder

Η οργάνωση του αποθηκευτικού χώρου 3/3

- **Κάθε έκδοση** ενός προγράμματος (V1, V2 κλπ.) θα είναι **ξεχωριστό project** με το δικό του ευρετήριο που θα βρίσκεται στο ευρετήριο της αντίστοιχης άσκησης. **Εναλλακτικά μπορείτε** να κρατάτε τις παλιές εκδόσεις στο ίδιο αρχείο σε σχόλια.

« My Drive > University > Ceid > I2P-LabExersices > HailStoneSequence

Name	Date modified	Type
HailStoneSequenceV1	13/10/2023 11:37	File folder
HailStoneSequenceV2	13/10/2023 11:47	File folder
HailStoneSequenceV3	13/10/2023 12:02	File folder

Οργάνωση Διάλεξης

- Τα παραδείγματα
 - MostFrequentNumber
 - AddNumbers
- Η οργάνωση του αποθηκευτικού χώρου
- **Διαδικασία Ανάπτυξης Προγράμματος**
 - Τα βήματα
 - Η Λεκτική Περιγραφή
 - Απλά Παραδείγματα (Hailstone, Fibonacci, Divisors)
- MostFrequentNumber - Διαδικασία Ανάπτυξης
 - Διαδοχικές εκδόσεις βελτίωσης του αλγορίθμου
- AddNumbers - Διαδικασία Ανάπτυξης
 - Incremental Development
- **Βασικά στοιχεία της γλώσσας C**

Διαδικασία Ανάπτυξης Προγράμματος

- **[E1]** Προσδιορίστε τις εισόδους και εξόδους της διεργασίας που απαιτείται για την εκτέλεση του έργου.
 - Τα βήματα της διεργασίας θα περιγράφει το πρόγραμμα σας.
- **[E2]** Προσδιορίστε ποιες από τις εισόδους είναι μεταβλητές και ποιες σταθερές.
- **[E3]** Δώστε τη λεκτική περιγραφή της διεργασίας αξιοποιώντας αφαιρετικότητα στις διεργασίες.
- **[E4]** Ελέγξτε αν εσείς, εκτελώντας τις ενέργειες που η λεκτική περιγραφή ορίζει, πετυχαίνετε τον στόχο της διεργασίας, για παράδειγμα τον προσδιορισμό του βαθμού του μαθήματος.
- **[E5]** Μετασχηματίστε τη λεκτική περιγραφή σε **C κώδικα**.
- **[E6]** Μετασχηματίστε τον πηγαίο κώδικα σε εκτελέσιμο.

Λεκτική περιγραφή - Ορισμός

- **είναι** μια περιγραφή, σε προστακτική μορφή, των ενεργειών που πρέπει να κάνει ένας άνθρωπος για να φέρει σε πέρας το έργο που του αναθέτουμε.
- **Περιγράφει** τη διεργασία που πρέπει να εκτελεστεί για να πάρουμε την έξοδο με βάση τις εισόδους.
- **Δημιουργείται πριν τη συγγραφή του κώδικα.**

Λεκτική περιγραφή – Ορισμός, Παραδείγμ.

Ορισμός

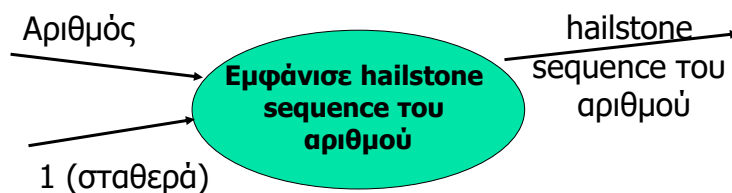
- [I2P_BasicConceptsPartA](#) (s6-7)
- I2P_ProgramDevelopmentUsingExamples (Current) ()

Παραδείγματα

- Add2Numbers
 - [I2P_BasicConceptsPartA](#) (s7,s16-18)
- HailstoneSequence
 - [I2P_BasicConceptsPartB](#) (s13-15)
- FibonacciNumbers
 - [I2P_BasicConceptsPartB](#) (s23)

HailstoneSequence – Ενέργειες E1, E2, E3

Αναπτύξτε ένα πρόγραμμα σύμφωνα με το οποίο το σύστημα θα δέχεται έναν αριθμό και θα εμφανίζει την hailstone sequence για τον αριθμό αυτό.

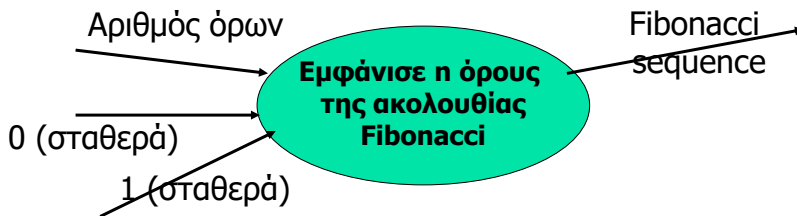


Go to partC

Δουλέψτε με τις λεκτικές περιγραφές που δόθηκαν στο [I2P_BasicConceptsPartB](#) (s11-15)

FibonacciNumbers – Ενέργειες E1, E2, E3

Αναπτύξτε ένα πρόγραμμα σύμφωνα με το οποίο το σύστημα θα δέχεται έναν αριθμό και θα εμφανίζει την hailstone sequence για τον αριθμό αυτό.



Δουλέψτε με τη λεκτική περιγραφή που δόθηκε στο [I2P_BasicConceptsPartB](#) (s23)

Divisors - Λεκτική περιγραφή

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

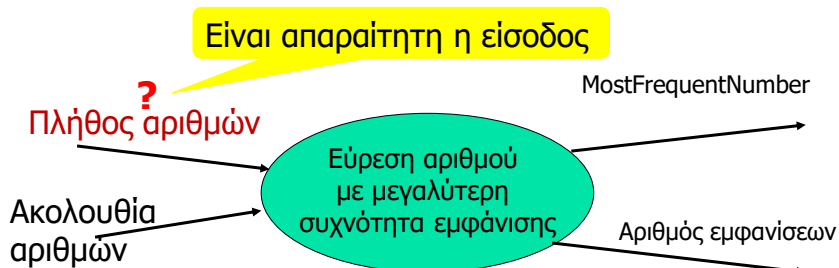
Οργάνωση Διάλεξης

- Τα παραδείγματα
 - MostFrequentNumber
 - AddNumbers
- Η οργάνωση του αποθηκευτικού χώρου
- Διαδικασία Ανάπτυξης Προγράμματος
 - Τα βήματα
 - Η Λεκτική Περιγραφή
 - Απλά Παραδείγματα (Hailstone, Fibonacci, Divisors)
- **MostFrequentNumber** - Διαδικασία Ανάπτυξης
 - Διαδοχικές εκδόσεις βελτίωσης του αλγορίθμου
- AddNumbers - Διαδικασία Ανάπτυξης
 - Incremental Development
- **Βασικά στοιχεία της γλώσσας C**

MostFrequentNumber – Ενέργειες E1, E2

Συχνότερα εμφανιζόμενος αριθμός

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



MostFrequentNumber–E3(Λεκτική Περιγραφή) 1/3

12 12 91 62 62 62 120 120 57

12 αριθμός

12 Υποψήφιος Συχνότερος Αριθμός (ΥΣΑ)

1 Πλήθος Εμφανίσεων (ΠΕ) ΥΣΑ

Πάρε τον πρώτο αριθμό
Θεώρησε τον αριθμό ΥΣΑ με ΠΕ 1

Για κάθε επόμενο αριθμό της ακολουθίας

■ **Αν** αυτός είναι ίσος με τον ΥΣΑ

- Αύξησε κατά 1 το ΠΕΥΣΑ

■ **Αλλιώς**

- Αντικατέστησε τον ΥΣΑ με τον αριθμό

Εμφάνισε Συχνότερο Αριθμό και Πλήθος Εμφανίσεων

Αναγνωσιμότητα
(Readability)
Απλότητα
(Simplicity)

Συνεχίστε τον έλεγχο
για διάφορα σενάρια
ακολουθίας

MostFrequentNumber–E3(Λεκτική Περιγραφή) 2/3

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

Εμφάνισε ΣΑ και ΠΕΣΑ

Βρείτε λάθη
Βελτιώστε τον

MostFrequentNumber–Ε3(Λεκτική Περιγραφή) 3/3

12 12 91 62 62 62 120 120 57

Πάρε τον πρώτο αριθμό

Θεώρησε τον αριθμό ΥΣΑ με ΠΕ 1

Για κάθε 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

Εμφάνισε ΣΑ και ΠΕΣΑ

Αναγνωσιμότητα
(Readability)
Απλότητα
(Simplicity)

Συνεχίστε τον έλεγχο
για διάφορα σενάρια
Ακολουθίας.
Βρείτε λάθη

Οργάνωση Διάλεξης

- Τα παραδείγματα
 - MostFrequentNumber
 - AddNumbers
- Η οργάνωση του αποθηκευτικού χώρου
- Διαδικασία Ανάπτυξης Προγράμματος
 - Τα βήματα
 - Η Λεκτική Περιγραφή
 - Απλά Παραδείγματα (Hailstone, Fibonacci, Divisors)
- MostFrequentNumber - Διαδικασία Ανάπτυξης
 - Διαδοχικές εκδόσεις βελτίωσης του αλγορίθμου
- **AddNumbers** - Διαδικασία Ανάπτυξης
 - Incremental Development του αλγορίθμου
- **Βασικά στοιχεία της γλώσσας C**

AddNumbers – Incremental Development

Γράψτε ένα πρόγραμμα σύμφωνα με το οποίο η μηχανή θα υπολογίζει το άθροισμα ακέραιων αριθμών. Προσέξτε τους περιορισμούς!

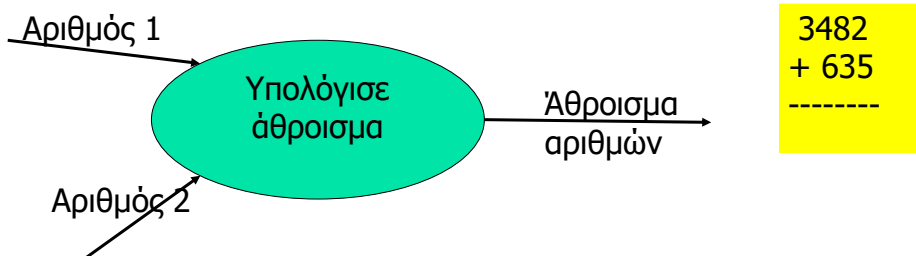
Περιορισμοί: Υποθέστε πως η μηχανή :

- a) δεν μπορεί να υπολογίσει άθροισμα αριθμών με περισσότερα από ένα ψηφία. Για τον υπολογισμό του αθροίσματος 2 ψηφίων χρησιμοποιήστε τον τελεστή +.
- b) μπορεί να κάνει τις πράξεις του πολλαπλασιασμού και διαίρεσης ακέραιων αριθμών με χρήση των τελεστών * και / αντίστοιχα. Επιπλέον μπορεί να βρίσκει το modulo ενός ακεραίου αριθμού με χρήση του τελεστή %.

Απαραίτητη η εφαρμογή της **Αυξητικής Ανάπτυξης**

- 1η έκδοση: Οι αριθμοί είναι 2 με 2 το πολύ ψηφία.
- 2^η έκδοση: Οι αριθμοί είναι 2 με 3 το πολύ ψηφία.
- 3^η έκδοση: Οι αριθμοί είναι 3 με 3 το πολύ ψηφία.
- 4^η έκδοση: Οι αριθμοί είναι k με n το πολύ ψηφία. **Αξιοποιήστε προτάσεις επανάληψης και πίνακες.**

AddNumbers – Ενέργειες E1, E2



- Επόμενο Βήμα: Λεκτική περιγραφή
- Λεκτική περιγραφή είναι η περιγραφή, σε προστακτική μορφή, των ενεργειών που πρέπει να κάνει ο **μαθητής της Α Δημοτικού** για να οδηγηθεί στο αποτέλεσμα με βάση τις εισόδους.

AddNumbers – Λεκτική περιγραφή

3482
+ 635

- Βάλε τον μεγαλύτερο αριθμό επάνω.
- Βάλε από κάτω τον άλλο αριθμό με στοίχιση στα δεξιά
- Ξεκίνα από δεξιά και για κάθε ψηφίο του πάνω αριθμού
 - Πρόσθεσε το με το ψηφίο του κάτω αριθμού
 - Αν υπάρχει από προηγούμενη πρόσθεση κρατούμενο
 - πρόσθεσε το
 - Αν το άθροισμα είναι μικρότερο από 9
 - Γράψε το από κάτω
 - Αλλιώς
 - Γράψε από κάτω το δεξιό ψηφίο του αθροίσματος
 - Κράτα ως κρατούμενο το αριστερό ψηφίο του αθροίσματος
- Δώσε το αποτέλεσμα της άθροισης.

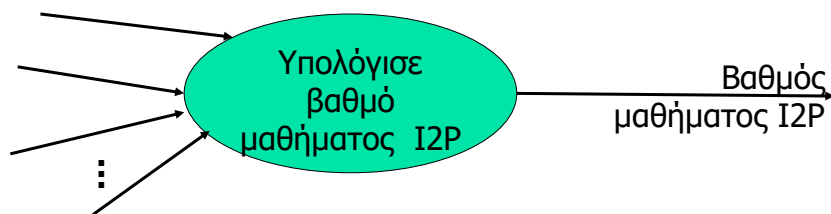
Προϋποθέτει:

A) γνώση του αλγορίθμου, και

B) δυνατότητα περιγραφής του στην ομιλούμενη γλώσσα

Ο πηγαίος κώδικας θα δημιουργηθεί στη συνέχεια με βάση την Λ.Π.

Η διεργασία υπολογισμού βαθμού μαθήματος I2P



- Είναι μια **σύνθετη** (παραγόμενη) **διεργασία** που η περιγραφή της θα δοθεί με τη χρήση άλλων διεργασιών
- Τι θα χρησιμοποιήσετε ως βάση;
- **Ποια διαδικασία θα ακολουθήσετε;**

Οργάνωση Διάλεξης

- Τα παραδείγματα
 - MostFrequentNumber
 - AddNumbers
- Η οργάνωση του αποθηκευτικού χώρου
- Διαδικασία Ανάπτυξης Προγράμματος
 - Τα βήματα
 - Η Λεκτική Περιγραφή
 - Απλά Παραδείγματα (Hailstone, Fibonacci, Divisors)
- MostFrequentNumber - Διαδικασία Ανάπτυξης
 - Διαδοχικές εκδόσεις βελτίωσης του αλγορίθμου
- AddNumbers - Διαδικασία Ανάπτυξης
 - Incremental Development του αλγορίθμου
- **Βασικά στοιχεία της γλώσσας C**

How I feel when my code works



Απαραίτητη Προϋπόθεση

- A) **Κατανόηση του προς ανάθεση έργου** (είσοδοι, έξοδοι, ...)
- B) **γνώση του αλγορίθμου,**
- Γ) **δυνατότητα περιγραφής του στην ομιλούμενη γλώσσα**
- Δ) δυνατότητα μετατροπής της λεκτικής περιγραφής σε source code
- E) παραγωγή εκτελέσιμου κώδικα

HailstoneSequence Εναλλακτικές

