

Εισαγωγή στον Προγραμματισμό

(CEID_NY131)



Βασικές έννοιες Διαδικαστικού Προστακτικού Προγραμματισμού (Μέρος 3^ο)
Η διεργασία ως πρόγραμμα και ως συνάρτηση

Kleanthis Thramboulidis
Prof. of Software and System Engineering
University of Patras

<https://sites.google.com/site/thramboulidiskleanthis/>



Βασικές έννοιες στην ανάπτυξη προγράμματος

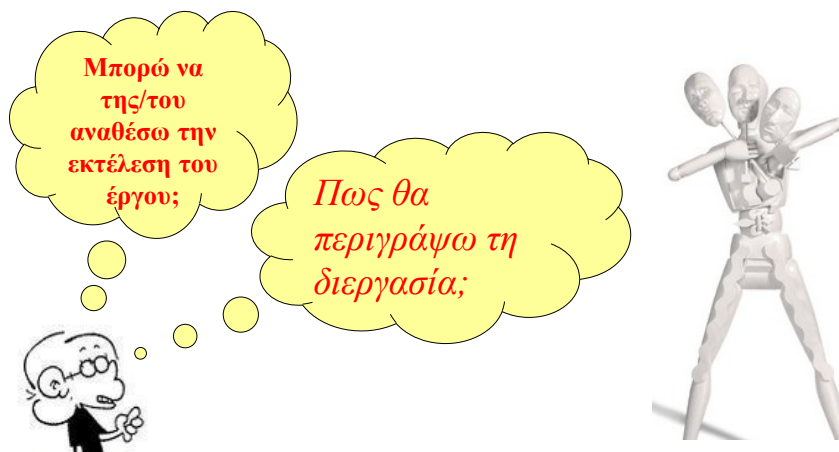
- Πρόγραμμα – Πηγαίος Κώδικας
- Συνάρτηση (Δήλωση, Κλήση, Ορισμός)
 - Η συνάρτηση **main()**
 - Συναρτήσεις της Βασικής Βιβλιοθήκης (printf(), scanf(), ...)
 - Δικές μας συναρτήσεις (getInt(), sum(), **average()**)
- Δεδομένα (Μεταβλητές) – Τύποι Δεδομένων (int) – **Πίνακας (Array)**
- Πρόταση
 - Πρόταση δήλωσης μεταβλητής
 - Πρόταση ελέγχου ροής προγράμματος (ελέγχου συνθήκης, επανάληψης)
- Τελεστής (+, =, <=, ++)
- Έκφραση (για επεξεργασία δεδομένων)

Οργάνωση Διάλεξης - Η επικοινωνία με την μηχανή

- **Η επικοινωνία με τη μηχανή**
 - Ο Τοτός και η πρόταση επανάληψης for
- Αφαιρετική Αναπαράσταση Διεργασίας (Fibonacci numbers)
 - Διαδικασία Ανάπτυξης Προγράμματος (Updated)
 - Εισαγωγή σε Πίνακες
- Add2Numbers (revisited)
 - Αφαιρετική αναπαράσταση – Εσείς - ο Τοτός και ... το ChatGPT
- Η διεργασία ως επαναχρησιμοποιήσιμη μονάδα κώδικα (συνάρτηση)
- Υπολογισμός μέσου όρου
 - Η συνάρτηση average (Πίνακας ως όρισμα σε συνάρτηση)

Η επικοινωνία με την μηχανή 1/2

Το έργο : "Θα γράψεις 500 φορές «Δεν θα πετάω αεροπλανάκια στην τάξη»"



Η επικοινωνία με την μηχανή 2/2

Γράψε μου την hailstone sequence για τον αριθμό 25

Ζήτα τον αριθμό
Πάρε τον αριθμό
....

Τι κάνουμε
τόρα;

Δεν καταλαβαίνω

Δεν καταλαβαίνω



Τεχνικές περιγραφής διεργασιών

- φυσική γλώσσα (Λεκτική περιγραφή)
- δομημένα αγγλικά (structured English)
- ψευδοκώδικας
- Διαγράμματα ροής (flow-charts)
- Program design languages
- SDL – STD – DFDs – ERDs – Petri Nets ...
- **UML** (Unified Modeling Language)
- **Γλώσσα Προγραμματισμού**

Τοτός – Μετατροπή σε πηγαίο κώδικα

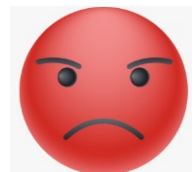
Εμφάνισε στην οθόνη το μήνυμα «Δεν θα ...»

Εμφάνισε στην οθόνη το μήνυμα «Δεν θα ...»

....

```
printf("I will not throw paper airplanes in class\n");  
printf("I will not throw paper airplanes in class\n");  
printf("I will not throw paper airplanes in class\n");  
printf("I will not throw paper airplanes in class\n");  
printf("I will not throw paper airplanes in class\n");  
printf("I will not throw paper airplanes in class\n");  
....
```

Πηγαίος Κώδικας



Η πρόταση επανάληψης for

Επανάλαβε 500 φορές
Εμφάνισε στην οθόνη το μήνυμα «Δεν θα ...»

Λεκτική Περιγραφή

```
for(count=1; count<=500; count++)  
    printf("I will not throw paper airplanes in class\n");
```

Πηγαίος Κώδικας

Η σημασία της πρότασης for

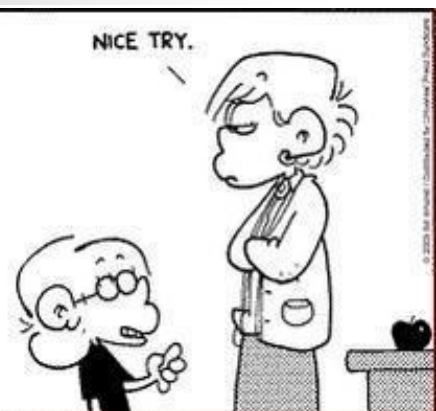
Για count ίσον 1 (count=1)
επανάλαβε όσο το count είναι μικρότερο ή ίσο με το 500 (count<=500)
printf(...)
αύξησε το count κατά 1 (count++ ↔ count = count+1)



Τα βασικά για προτάσεις ελέγχου ροής θα βρείτε στο [\(2P_ControlFlowStatements.pdf\)](#)

Τοτός – Εκτέλεση τιμωρίας

```
#include <stdio.h>  
int main(void)  
{  
    int count;  
    for (count = 1; count <= 500; count++)  
        printf("I will not throw paper airplanes in class.");  
    return 0;  
}
```



Περιγραφή διεργασίας σε C

```
#include <stdio.h>  Πρόταση Προ-επεξεργαστή
int main(void)     Ορισμός της συνάρτησης main
{
  Σηματοδοτεί την αρχή του σώματος της main
  int count;      Πρόταση δήλωσης μεταβλητής

  for (count = 1; count <= 500; count++) Loop Statement
    printf("I will not throw paper airplanes in class."); Function call Statement
  return 0;      Control flow Statement
}
Σηματοδοτεί το τέλος του σώματος της main
```

Οργάνωση Διάλεξης - Αφαιρετική Αναπαράσταση Διεργασίας

- Η επικοινωνία με τη μηχανή
 - Ο Τοτός και η πρόταση επανάληψης for
- **Αφαιρετική Αναπαράσταση Διεργασίας (Fibonacci numbers)**
 - Διαδικασία Ανάπτυξης Προγράμματος (Updated)
 - Εισαγωγή σε Πίνακες
- Add2Numbers (revisited)
 - Αφαιρετική αναπαράσταση – Εσείς - ο Τοτός και ... το ChatGPT
- Η διεργασία ως επαναχρησιμοποιήσιμη μονάδα κώδικα (συνάρτηση)
- Υπολογισμός μέσου όρου
 - Η συνάρτηση average (Πίνακας ως όρισμα σε συνάρτηση)

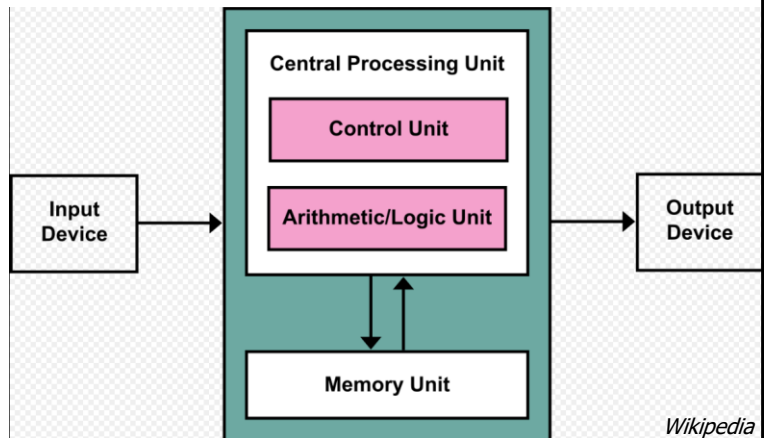
Ανάθεση έργου στη Μηχανή - Ερωτήσεις

- Τι πρέπει να διαθέτει το Robot για να του αναθέσω το έργο;
 - **computer hardware model**



- Ποια διαδικασία θα ακολουθήσω για την ανάθεση του έργου;
 - **development process**

Von Neumann architecture scheme



Wikipedia

Fibonacci sequence – Γλώσσα Μηχανής

- **32-bit x86 machine code**

```
8B542408 83FA0077 06B80000
0000C383 FA027706 B8010000
00C353BB 01000000 B9010000
008D0419 83FA0376 078BD98B
C84AEBF1 5BC3
```

8085 Microprocessor Trainer Kit With LED Display



1980



Fibonacci sequence – Assembly

- x86 assembly language using MASM syntax:

```
fib:
  mov edx, [esp+8]
  cmp edx, 0
  ja @f
  mov eax, 0
  ret

@@:
  lea eax, [ebx+ecx]
  cmp edx, 3
  jbe @f
  mov ebx, ecx
  mov ecx, eax
  dec edx
  jmp @b

@@:
  push ebx
  mov ebx, 1
  mov ecx, 1

@@:
  pop ebx
  ret
```

“An assembler is a computer program that translates assembly language code into machine code, allowing for direct communication with a computer's hardware. It converts human-readable instructions into **binary code** that the central processing unit (CPU) can execute.”

Source: <https://www.lenovo.com/>

Αφαιρετική αναπαράσταση - FibonacciNumbers

Αναπτύξτε ένα πρόγραμμα σύμφωνα με το οποίο το σύστημα θα εμφανίζει ζητούμενο αριθμό όρων της ακολουθίας Fibonacci.



Αναπαριστά τη διεργασία ως μια έλλειψη με το όνομα της και καταγράφει τις εισόδους και τις εξόδους της.

Μας βοηθά στον εντοπισμό των πρώτων μεταβλητών του προγράμματος.

Διαδικασία Ανάπτυξης Προγράμματος (Updated)

- **[Step1-(S1)]** Δώστε την **Αφαιρετική αναπαράσταση** της Διεργασίας.
Η Αφαιρετική αναπαράσταση **καταγράφει και α) τις εισόδους** που η διεργασία απαιτεί για την εκτέλεση του έργου **και β) τις εξόδους**.
 - Τα βήματα της διεργασίας θα περιγράφει η Λεκτική περιγραφή.
- **[S2] Δώστε τη λεκτική περιγραφή** της διεργασίας **αξιοποιώντας αφαιρετικότητα στις διεργασίες**.
 - **Ελέγξτε τη Λεκτική περιγραφή**
Ελέγξτε αν εσείς, εκτελώντας τις ενέργειες που η λεκτική περιγραφή ορίζει, πετυχαίνετε τον στόχο της διεργασίας, για παράδειγμα να δημιουργήσετε την ακολουθία Fibonacci.
- **[S3] Μετασχηματίστε τη λεκτική περιγραφή σε C κώδικα.**
Δηλώστε τις μεταβλητές που αντιστοιχούν σε εισόδους και εξόδους. Προσδιορίστε ποιες από τις εισόδους είναι μεταβλητές και ποιες σταθερές. Συνεχίστε με τις προτάσεις της Α.Π.
- **[S4] Μετασχηματίστε τον πηγαίο κώδικα σε εκτελέσιμο.**
- **[S5] Εκτελέστε και ελέγξτε το πρόγραμμα.**

Ο αναλυτικός έλεγχος μπορεί να αναβληθεί εν μέρη για το S5

Λεκτική περιγραφή - Ορισμός

- **είναι** μια δομημένη περιγραφή σε ομιλούμενη γλώσσα και σε **προστακτική μορφή** των ενεργειών που πρέπει να εκτελέσει μια οντότητα για να φέρει σε πέρας το έργο που της αναθέτουμε.
- **Περιγράφει** τη διεργασία που πρέπει να εκτελεστεί για να πάρουμε την έξοδο με βάση τις εισόδους.
- **Δημιουργείται πριν τη συγγραφή του κώδικα.**

Τι δεν περιλαμβάνει

- Περιγραφή ενεργειών του προγραμματιστή
- Περιγραφή ενεργειών του χρήστη
- Αναφορά σε μεταβλητές και γενικά κατασκευές της γλώσσας προγραμματισμού

Fibonacci sequence – [S2] Λεκτική περιγραφή

[S2] Δώστε τη λεκτική περιγραφή της διεργασίας αξιοποιώντας αφαιρετικότητα στις διεργασίες.

- Ελέγξτε τη Λεκτική περιγραφή

Ελέγξτε αν εσείς, εκτελώντας τις ενέργειες που η λεκτική περιγραφή ορίζει, πετυχαίνετε τον στόχο της διεργασίας, για παράδειγμα να δημιουργήσετε την ακολουθία Fibonacci.

Έργο: Τύπωσε την ακολουθία αριθμών Fibonacci

■ Λεκτική περιγραφή

- Πάρε τον αριθμό των όρων που ζητούνται
- Θεώρησε το 0 και 1 ως τους δύο πρώτους όρους
- Για όσους όρους ζητήθηκε
 - Υπολόγισε τον επόμενο ως το άθροισμα των 2 προηγούμενων του

1η εκδοχή

Θα έχουμε το αποτέλεσμα που θέλουμε με την 1^η αυτή εκδοχή;

Fibonacci sequence – [S2 και S3]

[S3] Μετασχηματίστε τη λεκτική περιγραφή σε **C κώδικα**.

Δηλώστε τις μεταβλητές που αντιστοιχούν σε εισόδους και εξόδους. Προσδιορίστε ποιες από τις εισόδους είναι μεταβλητές και ποιες σταθερές. Συνεχίστε με τις προτάσεις της Λ.Π.

Λεκτική περιγραφή

- Πάρε τον αριθμό των όρων που ζητούνται
- Πάρε τους δύο πρώτους όρους
- Εμφάνισε τους 2 πρώτους όρους
- Για όσους επιπλέον όρους ζητήθηκε
 - Υπολόγισε τον επόμενο ως το άθροισμα των 2 προηγούμενων του
 - Εμφάνισε τον τρέχοντα όρο
 - Ενημέρωσε τους δυο προηγούμενους όρους

1η εκδοχή

■ Ορισμός μεταβλητών

```
int numberOfTerms ;  
int fTerm, sTerm ;  
int cTerm ;
```

■ C κώδικας

Κάντε μια προσπάθεια να δώσετε τον πηγαίο κώδικα

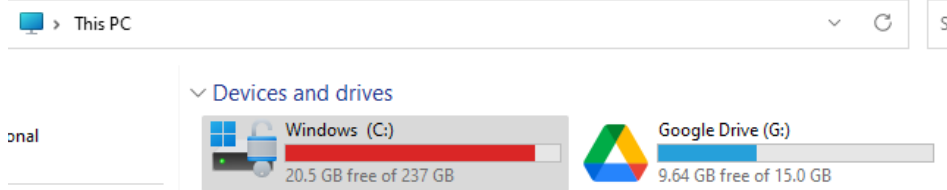
Δημιουργήστε ένα project στο DevCpp και γράψτε εκεί τον πηγαίο κώδικα. Μεταγλωττίστε και τρέξτε το πρόγραμμά σας. **Ελέγξτε τη λειτουργικότητά του!**

Switch to DevCpp

Η οργάνωση του αποθηκευτικού χώρου 1/2

Οδηγίες για την οργάνωση του αποθηκευτικού χώρου των ασκήσεων εργαστηρίου.

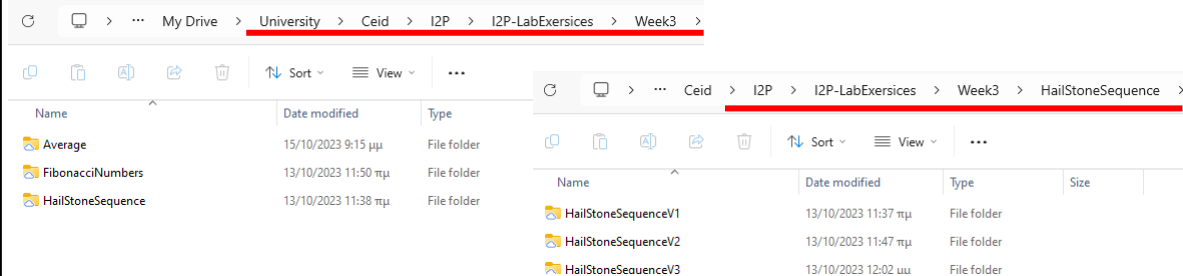
- **Ενιαίος αποθηκευτικός χώρος στο νέφος (Cloud)** [Οδηγίες αξιοποίησης Google drive από το IDE](#)
- Ενδεικτικό όνομα Ευρετηρίου: **I2P-LabExersices**



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

Η οργάνωση του αποθηκευτικού χώρου 2/2

- Κάθε άσκηση θα έχει το δικό της ευρετήριο, π.χ. Ευρετήριο HelloWorld, Ευρετήριο Add2Numbers
- **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.



Fibonacci sequence – C source code (V1)

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 // v1
4 int getInt(char message[]);
5
6 int numberOfTerms;
7 int fTerm=0, sTerm=1;
8 int cTerm;
9 int pTerm,ppTerm;
10
11 int main(int argc, char *argv[]) {
12     int i;
13     printf("FibonacciNumbers V1\n");
14     numberOfTerms=getInt("Dose arithmo orwn:");
15     printf("term-0:%d\n",fTerm);
16     printf("term-1:%d\n",sTerm);
17     ppTerm=fTerm;
18     pTerm=sTerm;
19     for(i=3;i<=numberOfTerms;i++){
20         cTerm=pTerm+ppTerm;
21         printf("term-%d:%d\n",i,cTerm);
22         ppTerm=pTerm;
23         pTerm=cTerm;
24     }
25     return 0;
26 }
```

Δήλωση και αρχικοποίηση Μεταβλητών

"Θεώρησε το 0 και 1 ως τους δύο πρώτους όρους"

"Πάρε τον αριθμό των όρων που ζητούνται"

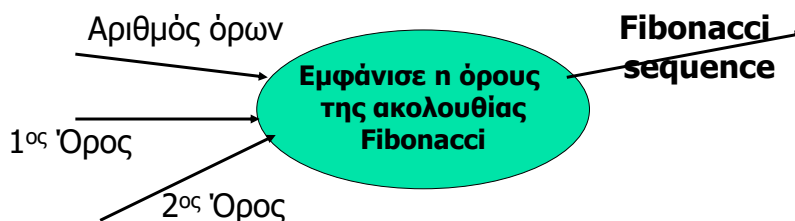
"Για όσους όρους ζητήθηκε"

- Υπολόγισε τον επόμενο ως το άθροισμα των 2 προηγούμενων του
- Εμφάνισε τον τρέχοντα όρο
- Ενημέρωσε τους δύο προηγούμενους όρους

Ελέγξτε τη συμπεριφορά του προγράμματος για διάφορα σενάρια

FibonacciNumbers V2 – Αναπαράσταση Εξόδου

Αναπτύξτε ένα πρόγραμμα σύμφωνα με το οποίο το σύστημα θα εμφανίζει ζητούμενο αριθμό όρων της ακολουθίας Fibonacci.

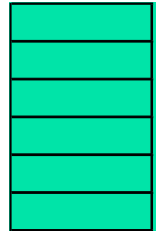


Δηλώστε τις κατάλληλες μεταβλητές για την αποθήκευση της εξόδου του προγράμματος

Μήπως αυτό επηρεάζει τη Λεκτική Περιγραφή (δηλαδή τον Αλγόριθμο) ;

Πίνακας (Array) []

- **χρησιμοποιείται**
 - για αποθήκευση και διαχείριση μεγάλων ποσοτήτων δεδομένων που είναι **κοινού τύπου** και **σχετίζονται μεταξύ τους**.
- **για την γλώσσα είναι**
 - μια συλλογή στοιχείων ίδιου τύπου, των οποίων οι τιμές είναι αποθηκευμένες σε διαδοχικές θέσεις μνήμης
- **αποτελούν**
 - μαζί με τους δείκτες από τα πιο ισχυρά στοιχεία της C

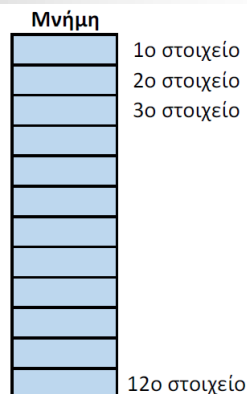


Τα βασικά πίνακες θα βρείτε στο [l2P_Arrays.pdf](#)

Δήλωση Πίνακα

- **δηλώνει στον compiler**
 - το όνομα του πίνακα (fbn)
 - τον αριθμό των στοιχείων του (12)
 - τον τύπο των στοιχείων του (int)

```
int fbn[12];
```



Το μέγεθος του πίνακα δεν μπορεί να αλλάξει στο χρόνο εκτέλεσης του προγράμματος (run-time).

Fibonacci sequence – C source code (V2)

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 // V2
4 int getInt(char message[]);
5
6 int numberOfTerms;
7
8 int main(int argc, char *argv[]) {
9     int i;
10    printf("FibonacciNumbers V1\n");
11    numberOfTerms=getInt("Dose arithmo oron:");
12
13    int fbNumber[numberOfTerms];
14
15    fbNumber[0]=getInt("Dose 1o oro:");
16    fbNumber[1]=getInt("Dose 2o oro:");
17
18    for(i=2;i<=numberOfTerms;i++){
19        fbNumber[i]=fbNumber[i-1]+fbNumber[i-2];
20    }
21    for(i=0;i<numberOfTerms;i++){
22        printf("term-%d:%d\n", i+1, fbNumber[i]);
23    }
24    return 0;
}
```

Η δήλωση του πίνακα fbNumber πρέπει να γίνει μετά την απόδοση τιμής στην μεταβλητή numberOfTerms για να δεσμευθεί ο απαραίτητος χώρος στη μνήμη.

Για όσους επιπλέον όρους ζητήθηκε

- Υπολόγισε τον επόμενο ως το άθροισμα των 2 προηγούμενων του

Εμφάνισε τους όρους της ακολουθίας

Οργάνωση Διάλεξης - Add2Numbers (revisited)

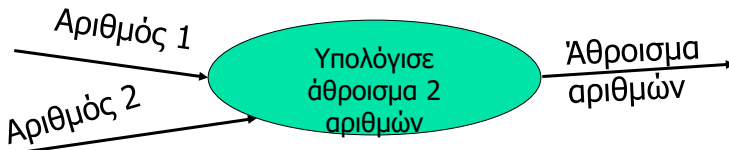
- Η επικοινωνία με τη μηχανή
 - Ο Τοτός και η πρόταση επανάληψης for
- Αφαιρετική Αναπαράσταση Διεργασίας (Fibonacci numbers)
 - Διαδικασία Ανάπτυξης Προγράμματος (Updated)
 - Εισαγωγή σε Πίνακες
- **Add2Numbers (revisited)**
 - Αφαιρετική αναπαράσταση – Εσείς - ο Τοτός και ... το ChatGPT
- Η διεργασία ως επαναχρησιμοποιήσιμη μονάδα κώδικα (συνάρτηση)
- Υπολογισμός μέσου όρου
 - Η συνάρτηση average (Πίνακας ως όρισμα σε συνάρτηση)

Add2Numbers – [S1, S2] Αφαιρετική αναπαράσταση

[Step1-(S1)] Δώστε την **Αφαιρετική αναπαράσταση** της Διεργασίας.

Η Αφαιρετική αναπαράσταση **καταγράφει και α) τις εισόδους** που η διεργασία απαιτεί για την εκτέλεση του έργου **και β) τις εξόδους**.

- Τα βήματα της διεργασίας θα περιγράφει η λεκτική περιγραφή. **[S2] Δώστε τη λεκτική περιγραφή** της διεργασίας **αξιοποιώντας αφαιρετικότητα στις διεργασίες.**



Η διεργασία Υπολόγισε άθροισμα είναι μια **σύνθετη** (παραγόμενη) **διεργασία** που η περιγραφή της δίνεται με την χρήση άλλων διεργασιών

Λεκτική Περιγραφή της Διεργασίας “Υπολόγισε Άθροισμα 2 αριθμών”

- Πάρε τον 1^ο αριθμό
- Πάρε τον 2^ο αριθμό
- Πρόσθεσε αριθμούς
- εμφάνισε αποτέλεσμα

Add2Numbers – [S3]

[S3] Μετασχηματίστε τη λεκτική περιγραφή σε **C κώδικα**.

Δηλώστε τις μεταβλητές που αντιστοιχούν σε εισόδους και εξόδους. Προσδιορίστε ποιες από τις εισόδους είναι μεταβλητές και ποιες σταθερές. Συνεχίστε με τις προτάσεις της Λ.Π.

■ Ορισμός μεταβλητών

```
int num1, num2;  
int sum;
```

■ C κώδικας

Λεκτική περιγραφή

- Πάρε τον 1^ο αριθμό — Παραγόμενη διεργασία. Περιγράφεται από **δική μας συνάρτηση (getInt())**
- Πάρε τον 2^ο αριθμό — Παραγόμενη διεργασία. Περιγράφεται από **δική μας συνάρτηση (getInt())**
- Πρόσθεσε αριθμούς — Πρωτογενής/Βασική διεργασία. Περιγράφεται από **τελεστή (+) της Γλώσσας**
- εμφάνισε αποτέλεσμα — Παραγόμενη διεργασία. Περιγράφεται από **συνάρτηση της Βασικής βιβλιοθήκης (printf())**

Add2Numbers [S1-S5] – Εσείς

Εργαστήριο 2ης εβδομάδας (Αντικείμενο)

Δευτέρα, 7 Οκτωβρίου 2024 - 9:31 π.μ. - από τον χρήστη Θραμπουλίδης Κλεάνθης

A) Δράσεις

Θα εκτελέσετε τις παρακάτω Δράσεις με την σειρά που παρατίθενται.

[Δ1] - ολοκλήρωση του Εργαστηρίου της 1ης Εβδομάδας

Θα πρέπει να ολοκληρώσετε επιτυχώς τις δράσεις του Εργαστηρίου της 1^{ης} Εβδομάδας. Χωρίς αυτό δεν έχει νόημα να πάτε στις ει

[Δ2] - Ανάπτυξη 1ης έκδοσης του προγράμματος Add2Numbers

Αναπτύξτε την 1^η έκδοση του προγράμματος Add2Numbers αξιοποιώντας μόνο συναρτήσεις της βασικής βιβλιοθήκης (printf) ar έκδοση αυτή δεν χρησιμοποιούμε δική μας συνάρτηση.

Για να έχει καλύτερο αποτέλεσμα για εσάς η διαδικασία προσπαθήστε να αναπτύξετε το πρόγραμμα αφού μελετήσετε πρώτα τις δ να κάνετε copy paste από τον κώδικα που αναπτύξαμε στην διάλεξη. Αν χρειαστεί ανατρέξτε να λύσετε το πρόβλημα που θα συναντήσετε πάλι γράφοντας μόνοι σας κώδικα.

[Δ3] - Ανάπτυξη 2ης έκδοσης του προγράμματος Add2Numbers

Για να έχει καλύτερο αποτέλεσμα για εσάς η διαδικασία προσπαθήστε να αναπτύξετε το πρόγραμμα αφού μελετήσετε πρώτα τις διαφάνειες και χωρίς να κάνετε copy paste από τον κώδικα που αναπτύξαμε στην διάλεξη. Αν χρειαστεί ανατρέξτε να λύσετε το πρόβλημα που θα συναντήσετε και συνεχίστε πάλι γράφοντας μόνοι σας κώδικα. (Εργαστήριο 2ης εβδομάδας (Αντικείμενο))

Γ) Δράσεις για μετά την ολοκλήρωση των [Δ1]-[Δ4]

Βελτιώσεις για μια καλύτερη εκδοχή της getNumber

[Δ5] - Βελτίωση που αφορά το όνομα της συνάρτησης

Τώρα που ξέρουμε πως υπάρχουν και αριθμοί κινητής υποδιαστολής (double) μπορούμε (και είναι καλύτερο) να δώσουμε ως όνομα σ αργότερα (σε άλλη άσκηση που πιθανόν να χρειάζεται) άλλες δύ

[Δ6] - Βελτίωση που αφορά τη λειτουργικότητα

Το αναπτύξαμε στη Διάλεξη. Το είχατε διαθέσιμο στο eclass ([Add2NumbersV1.pdf](#), [Add2NumbersV2.pdf](#)).

Παραδοτέο

Ασκήσεις

85 Υποβολές

Ερωτηματολόγιο

59 Συμμετοχές

Add2Numbers και Τοτός

Write a program in C to take 2 numbers and display the sum

Θα γράψεις ένα πρόγραμμα που θα παίρνει 2 αριθμούς και θα δίνει το άθροισμα

```
Certainly! Below is a simple C program that takes two numbers as input from the user and displays their sum:

c Copy code

#include <stdio.h>

int main() {
    int num1, num2, sum;

    // Take input from the user
    printf("Enter first number: ");
    scanf("%d", &num1);

    printf("Enter second number: ");
    scanf("%d", &num2);

    // Calculate the sum
    sum = num1 + num2;

    // Display the sum
    printf("Sum: %d\n", sum);

    return 0;
}
```



Add2Numbers και ChatGPT



Write a program in C to take 2 numbers and display the sum



Certainly! Below is a simple C program that takes two numbers as input and displays their sum:

ChatGPT

```
c Copy code
#include <stdio.h>

int main() {
```



GPT-3.5

<https://chat.openai.com/share/dd992023-516f-4d31-ad5d-71459f052eb2/dd992023-516f-4d31-ad5d-71459f052eb2>

Here's how the program works:

1. It includes the necessary header file `stdio.h` for standard input/output operations.
2. `main()` is the entry point of the program.
3. It declares three integer variables: `num1`, `num2`, and `sum`.
4. The program prompts the user to enter the first and second numbers using `printf` and reads the input using `scanf`.
5. It calculates the sum of `num1` and `num2` and stores the result in the variable `sum`.
6. Finally, it prints out the result using `printf`.

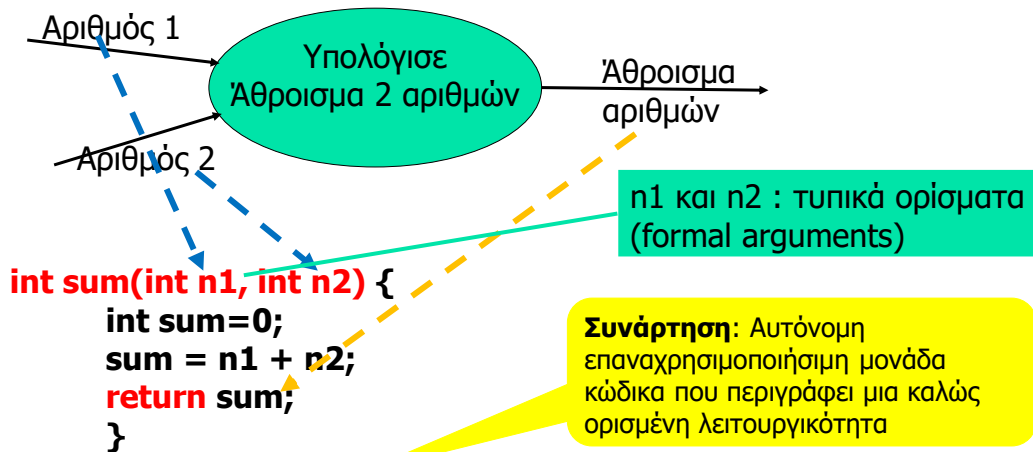
You can compile and run this program using a C compiler. When you run the program, it will ask you to enter two numbers, and then it will display their sum.

Οργάνωση Διάλεξης - Η διεργασία ως συνάρτηση

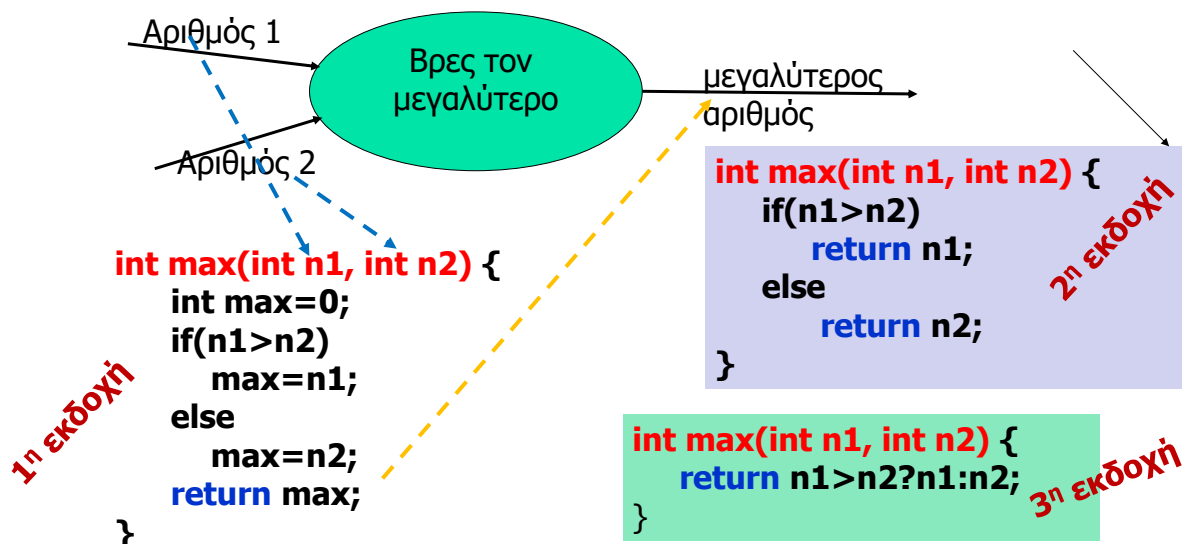
- Η επικοινωνία με τη μηχανή
 - Ο Τοτός και η πρόταση επανάληψης for
- Αφαιρετική Αναπαράσταση Διεργασίας (Fibonacci numbers)
 - Διαδικασία Ανάπτυξης Προγράμματος (Updated)
 - Εισαγωγή σε Πίνακες
- Add2Numbers (revisited)
 - Αφαιρετική αναπαράσταση – Εσείς - ο Τοτός και ... το ChatGPT
- **Η διεργασία ως επαναχρησιμοποιήσιμη μονάδα κώδικα (συνάρτηση)**
- Υπολογισμός μέσου όρου
 - Η συνάρτηση average (Πίνακας ως όρισμα σε συνάρτηση)

Διεργασία ως Reusable function - Η συνάρτηση sum

Η διεργασία ως αυτόνομη επαναχρησιμοποιήσιμη μονάδα λειτουργικότητας



Διεργασία ως Reusable function - Η συνάρτηση max



Λεκτική περιγραφή => Πηγαίος κώδικας (if else και for)

Έλεγχος συνθήκης

Εάν a μεγαλύτερο από b
τότε το μέγιστο είναι το a
αλλιώς
το μέγιστο είναι το b



```
if(a>b)
    max=a;
else
    max=b;
```

Επανάληψη

Για 500 φορές
εμφάνισε «Δεν θα πετάω αεροπλανάκια ...»



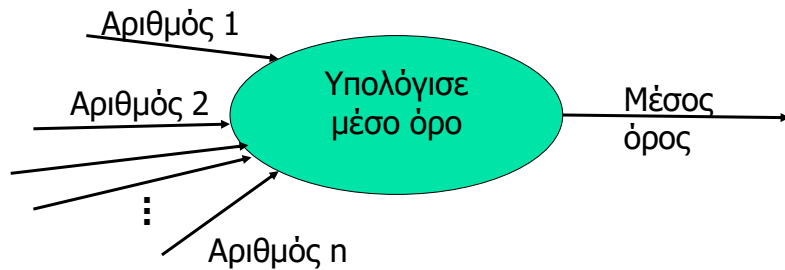
```
for (count = 1; count <= 500; count++)
    printf("I will not throw paper airplanes in class.");
```

Οργάνωση Διάλεξης - Υπολογισμός μέσου όρου

- Η επικοινωνία με τη μηχανή
 - Ο Τοτός και η πρόταση επανάληψης for
- Αφαιρετική Αναπαράσταση Διεργασίας (Fibonacci numbers)
 - Διαδικασία Ανάπτυξης Προγράμματος (Updated)
 - Εισαγωγή σε Πίνακες
- Add2Numbers (revisited)
 - Αφαιρετική αναπαράσταση – Εσείς - ο Τοτός και ... το ChatGPT
- Η διεργασία ως επαναχρησιμοποιήσιμη μονάδα κώδικα (συνάρτηση)
- **Υπολογισμός μέσου όρου**
 - Η συνάρτηση average (**Πίνακας ως όρισμα σε συνάρτηση**)

Η διεργασία υπολογισμού μέσου όρου ενός συνόλου ακεραίων

Το έργο : Υπολογισμός μέσου όρου ενός συνόλου ακεραίων



- Είναι μια **σύνθετη** (παραγόμενη) **διεργασία** που η περιγραφή της δίνεται με την χρήση άλλων διεργασιών

Πόσες μεταβλητές χρειάζεστε για τα δεδομένα εισόδου;

Μήπως εξαρτάται από την Λεκτική Περιγραφή (δηλαδή τον Αλγόριθμο) ;

Average ως πρόγραμμα – Λεκτική Περιγραφή

Λεκτική περιγραφή No 1 (V1)

1. Πάρε το πλήθος των αριθμών
2. **Για το πλήθος των αριθμών επανέλαβε**
 1. Πάρε αριθμό
 2. Ενημέρωσε το άθροισμα
3. Υπολόγισε μέσο όρο
4. Εμφάνισε το μέσο όρο

Switch to DevCpp

Μετατροπή σε πηγαίο κώδικα

Προτάσεις επανάληψης (iteration statements)

- **for()**
- **while()**
- **do while()**

Τα βασικά για προτάσεις ελέγχου ροής θα βρείτε στο [\(I2P_ControlFlowStatements.pdf\)](#)

Average με for (V1)

float: τύπος δεδομένων που προσδιορίζει αριθμούς κινητής υποδιαστολής απλή ακρίβειας

Γιατί όχι int avg

Πρόταση Επανάληψης for

τρέξτε το πρόγραμμα για τους αριθμούς 2 και 3

```
calcAverageV1.c
1 #include <stdio.h>
2 #include <stdlib.h>
3 int getInt(char message[]);
4
5 int nunOfTerms, num, sum;
6 float avg;
7
8 int main(int argc, char *argv[]) {
9     int i=0;
10
11     printf("calcAverage with while\n");
12     nunOfTerms = getInt("Dose to plithos tw n arithmw n:");
13     for(i=0;i<nunOfTerms;i++){
14         num= getInt("Dose arithmo:");
15         sum += num;
16     }
17     avg = sum/nunOfTerms;
18     printf("Average is %f\n", avg);
19     return 0;
20 }
21
22 int getInt(char message[]){
```

Προσέξτε τον προσδιοριστή %f

Average με while (V1)

float: τύπος δεδομένων που προσδιορίζει αριθμούς κινητής υποδιαστολής απλή ακρίβειας

Πρόταση επανάληψης

τρέξτε το πρόγραμμα για τους αριθμούς 2 και 3

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int numofNumbers, count, num, sum;
5 float avg;
6
7 int main(int argc, char *argv[]) {
8
9     printf("CalcAverage using while (v1)\n");
10    printf("Dose plithos arithmw n:");
11    scanf("%d",&numofNumbers);
12    // printf("%d\n",numofNumbers );
13    count=0;
14    while(count<numofNumbers){
15        printf("Dose arithmo:");
16        scanf("%d",&num);
17        sum=sum+num; // sum+=num;
18        count = count+1; //count++;
19    }
20    printf("sum=%d\n",sum );
21    avg = sum/numofNumbers;
22    printf("avg=%f", avg );
23    return 0;
24 }
```

CalcAverageUsingWhileV1

Average με for (V2)

τρέξετε το πρόγραμμα για τους αριθμούς 2 και 3

τρέξετε το πρόγραμμα για πλήθος αριθμών 0

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int numOfNumbers, count, num, sum;
5 float avg;
6
7 int main(int argc, char *argv[]) {
8     printf("CalcAverage using for (v1)\n");
9     printf("Dose plithos arithmwn:");
10    scanf("%d",&numOfNumbers);
11    // printf("%d\n",numOfNumbers );
12    for(count=0;count<numOfNumbers;count++){
13        printf("Dose arithmo:");
14        scanf("%d",&num);
15        sum=sum+num; // sum+=num;
16    }
17    // printf("sum=%d\n",sum );
18    avg = (float)sum/numOfNumbers;
19    printf("avg=%f",avg );
20    return 0;
21 }
```

CalcAverageUsingForV1

Κλεάνθης Θραμπουλίδης

Βασικές Έννοιες (Μέρος 2)

43

Average (V3)

Πρόταση **Ελέγχου Συνθήκης**. Το σώμα της if θα εκτελεστεί αν η έκφραση στην παρένθεση έχει τιμή αληθη (true)

(float)sum: Μετατροπή τύπου (type conversion). Μετατρέπει το sum από int σε float. Έτσι το σύστημα αναγκάζεται να κάνει διαίρεση με αριθμούς κινητής υποδιαστολής.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int numOfNumbers, count, num, sum;
5 float avg;
6
7 int main(int argc, char *argv[]) {
8     printf("CalcAverage using for (v1)\n");
9     printf("Dose plithos arithmwn:");
10    scanf("%d",&numOfNumbers);
11    // printf("%d\n",numOfNumbers );
12    if(numOfNumbers==0){
13        printf("No average for 0 numbers\n");
14        return 0;
15    }
16    for(count=0;count<numOfNumbers;count++){
17        printf("Dose arithmo:");
18        scanf("%d",&num);
19        sum=sum+num; // sum+=num;
20    }
21    // printf("sum=%d\n",sum );
22    avg = (float)sum/numOfNumbers;
23    printf("avg=%f",avg );
24    return 0;
25 }
```

CalcAverageUsingForV2

Κλεάνθης Θραμπουλίδης

Βασικές Έννοιες (Μέρος 3^ο)

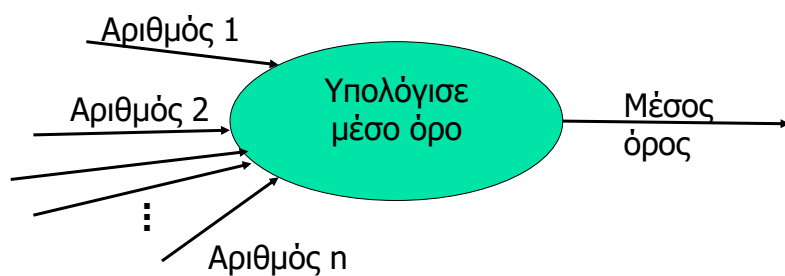
44

for vs. while

```
13 count=0;
14 while(count<nunOfNumbers){
15     scanf("%d", &num);
16     sum += num;
17     count++;
18 }

12 for(count=0;count<nunOfNumbers;count++){
13     scanf("%d", &num);
14     sum += num;
15 }
```

Average ως Συνάρτηση

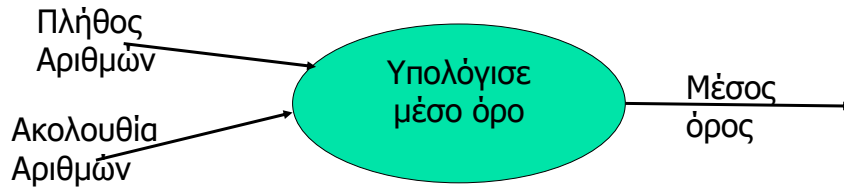


?

```
float average(int n1, int n2, int n3, int n4, int n5) {
    int sum=0;
    sum = n1+n2+n3+n4+n5;
    return sum/5;
}
```

Είναι reusable η συνάρτηση;

Average ως Συνάρτηση με όρισμα Πίνακα



```
float average(int num[], int numOfNumbers) {  
    int i;  
    int sum=0;  
    for(i=0;i<numOfNumbers;i++)  
        sum+=n[i];  
    return (float)sum/numOfNumbers;  
}
```

Average με function (V4) 1/2

```
Project Classes Debug  
CalcAverageUsingFunctionV1  
main.c  
1 #include <stdio.h>  
2 #include <stdlib.h>  
3  
4 float average(int n[],int s);  
5  
6 int numOfNumbers, count, sum;  
7 int num[100];  
8 float avg;  
9  
10 int main(int argc, char *argv[]) {  
11     printf("Dose plithos arithmwn:");  
12     scanf("%d",&numOfNumbers);  
13     for(count=0;count<numOfNumbers;count++){  
14         printf("Dose %d arithmo:",count+1);  
15         scanf("%d",&num[count]);  
16     }  
17  
18     avg = average(num,numOfNumbers);  
19  
20     printf("avg=%f",avg );  
21     return 0;  
22 }  
23  
24 float average(int n[],int s){  
25     int sum=0;
```


Average με function (V4) 2/2

```
18     avg = average(num,numOfNumbers);
19
20     printf("avg=%f",avg );
21     return 0;
22 }
23
24 float average(int n[],int s){
25     int sum=0;
26     int i;
27
28     for(i=0;i<s;i++)
29         sum+=n[i];
30     return (float)sum/s;
31 }
```

Η άσκηση AddNumbers (για προχωρημένους)

Η διεργασία πρόσθεσης αριθμών

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

Περιορισμοί:

Υποθέστε πως η μηχανή :

- δεν μπορεί να υπολογίσει άθροισμα αριθμών με περισσότερα από ένα ψηφία. Για τον υπολογισμό του αθροίσματος 2 ψηφίων χρησιμοποιήστε τον τελεστή +.
- 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 %.

Αξιοποιήστε την **αυξητική ανάπτυξη**.

Αναπτύξτε πρώτα μια έκδοση για αριθμούς με 2 ψηφία.
Στη συνέχεια μια έκδοση για αριθμούς με 3 ψηφία.

How I feel when my code works

