

# Εισαγωγή στην Πληροφορική & στον Προγραμματισμό

---

Αρχές Προγραμματισμού Η/Υ (με τη γλώσσα C)

Διάλεξη #3

Παναγιώτης Παύλου

Παρασκευή, 19 Μαρτίου  2021

[c-programming@mail.ntua.gr](mailto:c-programming@mail.ntua.gr)

# Πρακτική

---

Εργαστηριακά παραδείγματα

# Πρακτική #1

---



Οι συναρτήσεις που επιστρέφουν λογικές τιμές συνήθως ονομάζονται με όνομα που ξεκινά το `is` ώστε να “ρωτά” για το αν υπάρχει ή αν ισχύει κάτι.

Ας γράφει μία τέτοια συνάρτηση που να επιστρέφει εάν ένας ακέραιος είναι ζυγός.

```
bool isEven(int N)
```

# Πρακτική #2

---



Το ζητούμενο σε αυτές τις εφαρμογές είναι να δημιουργηθούν συναρτήσεις:

1. Την `maxOf2` που επιστρέφει τη μεγαλύτερη από δύο τιμές A, B

```
double maxOf2 (double A, double B)
```

2. Την `maxOf3` που επιστρέφει τη μεγαλύτερη από τρεις τιμές A, B, C:

```
double maxOf3 (double A, double B , double C)
```

# Πρακτική #3



Για ζέσταμα ας ξεκινήσουμε με έναν κώδικα που να επιστρέφει το τελευταίο ψηφίο (αυτό των μονάδων) ενός αριθμού  $N$  όταν αυτός γραφεί σε ένα σύστημα αρίθμησης με βάση  $B$ . Η βάση επιτρέπεται να είναι 8, 10 ή 16. Αν δεν είναι, τότε επιστρέφεται η τιμή -1 ως ένδειξη λάθους.

```
int lastDigit(unsigned int N, int B)
```

Για παράδειγμα:

```
lastDigit( 123, 10) → 3
```

```
lastDigit(0x1F, 16) → 15 (την τιμή του F)
```

```
lastDigit( 046, 8) → 6
```

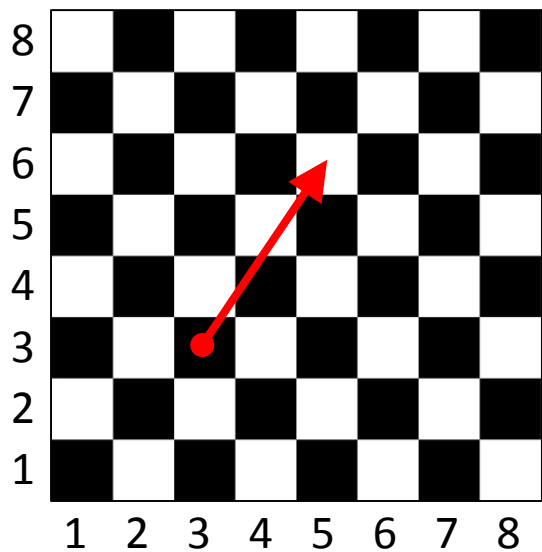
```
lastDigit( 56, 16) → 8 (επειδή ο αριθμός είναι γραμμένος στο δεκαδικό)
```

# Πρακτική #4



Ας δούμε μία συνάρτηση που επιστρέφει το κατά πόσο επιτρέπεται να γίνει μία κίνηση στο σκάκι (και υποθέτουμε άδεια σκακιέρα).

```
bool isValidMove(int piece, int row1, int col1, int row2, int col2)
```



Η πρώτη παράμετρος είναι το **κομμάτι** που θα κινηθεί, οι δύο επόμενες η γραμμή και η στήλη της σκακιέρας που ξεκινά το κομμάτι και οι δύο τελευταίες η γραμμή και η στήλη της σκακιέρας που καταλήγει το κομμάτι.

Για να ονοματίσουμε τα κομμάτια ορίζουμε/κωδικοποιούμε τις αντιστοιχίσεις σταθερών:  
**1:**Πιόνι , **2:**Αξιωματικός, **3:**Ίππος, **4:**Πύργος,  
**5:**Βασίλισσα, **6:** Βασιλιάς

# Σημαντικά σημεία



Μετά από τη σημερινή διάλεξη θα πρέπει να γνωρίζετε:

- Τι είναι η κλήση με τιμή (call-by-value)
- Ποια είναι η εμβέλεια των τοπικών μεταβλητών
- Τι είναι οι λογικές μεταβλητές και οι λογικές προτάσεις
- Τους τελεστές σύγκρισης και τη χρήση τους
- Τους λογικούς τελεστές (bool) και τη χρήση τους
- Την εντολή if/else if/else και τη χρήση της
- Τον τριαδικό τελεστή
- Την πολλαπλή ανάθεση τιμής
- Τι είναι το smProject και γιατί το χρησιμοποιούμε

# Πρακτικά ζητήματα

---

smProject – Γιατί σας βοηθά να παραδώσετε σωστότερο αποτέλεσμα



# Χρήση smProject

---

Η κάθε εργασία, εκτός από την εκφώνησή της, θα συνοδεύεται και από ένα CLion project το οποίο έχει σαν σκοπό να σας βοηθά να παραδίδετε πιο σωστές εργασίες, επειδή:

1. Περιλαμβάνει κάποιες δοκιμές για κάποια συνηθισμένα σφάλματα πάνω στα ζητούμενα. Έτσι θα έχετε την ευκαιρία να δείτε μόνοι σας εάν ο κώδικάς σας έχει κάποιο τέτοιο σφάλμα.  
Η λίστα των ελέγχων είναι ενδεικτική και όχι εξαντλητική, έτσι εάν πετύχουν όλοι οι έλεγχοι, πιθανώς να εξακολουθούν να υπάρχουν αρκετά και σοβαρά λογικά σφάλματα στον κώδικά σας. Και τα σταματημένα ρολόγια 2 φορές τη μέρα δείχνουν τη σωστή ώρα.
2. Έχει έτοιμες τις δηλώσεις των συναρτήσεων που ζητούνται, οπότε έχετε μία ευκολία παραπάνω. Σωστό όνομα συνάρτησης, σωστές παραμέτρους και τύπο δεδομένων του αποτελέσματος.
3. Έχει οριοθετημένο τον κώδικα με ένα σχόλιο στην αρχή και ένα στο τέλος. Εφόσον εσείς γράψετε όλο τον κώδικά σας ανάμεσα σε αυτά τα σχόλια, τότε σε συνδυασμό με το αναβαθμισμένο σύστημα υποβολής, εάν δεν έχετε αντιγράψει ολόκληρο τον κώδικα θα εμφανιστεί μία ειδοποίηση γι' αυτό.

**Από τα παραπάνω είναι προφανές ότι για κάθε άσκηση θα πρέπει να κατεβάζετε το αντίστοιχο smProject αυτής της άσκησης και σε αυτό το project να γράφετε τον κώδικα.**

Για να είναι δυνατή η λειτουργία του smProject, χρειάζεται η παραδοχή ότι αντί της `main`, η κύρια συνάρτηση του προγράμματος θα είναι η `smMain` με την ίδια ακριβώς σύνταξη που έχει η `main`.

Στην επόμενη διαφάνεια θα δείτε πως, όταν ανοίξετε το smProject στο CLion, θα επιλέγετε την εκτέλεση της κανονικής λειτουργίας ή των δοκιμαστικών ελέγχων.

# Κανονική εκτέλεση και δοκιμές

Στην γραμμή εκτέλεσης εμφανίζονται πλέον δύο **στόχοι**.

- **smProject** , που αφορά την κανονική εκτέλεση του κώδικα
- **RunTests** , που αφορά την εκτέλεση των δοκιμών

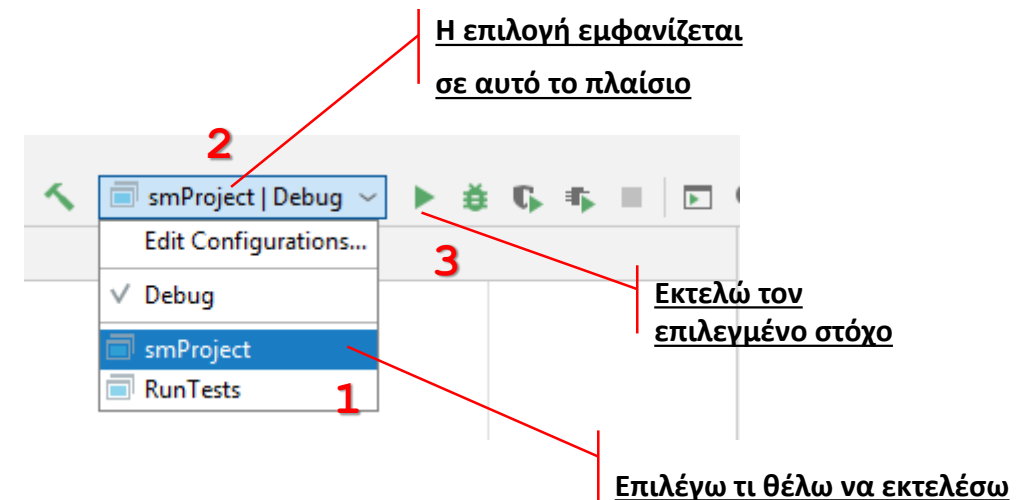
Θα πρέπει λοιπόν ο χρήστης, ανάλογα με την επιθυμία του, να επιλέξει τον αντίστοιχο στόχο.

Αφού έχει γίνει η επιλογή αυτή τα υπόλοιπα κουμπιά της γραμμής λειτουργούν κατά τα γνωστά.

Ως συνήθης πρακτική προτείνεται να γίνονται τα πάντα σε κανονική εκτέλεση μέχρι να θεωρήσει ο προγραμματιστής ότι ο κώδικας είναι έτοιμος.

Κατόπιν να γίνεται αλλαγή του στόχου σε RunTests για να βεβαιωθεί ότι όλα εκτελούνται καλά.

Ενδεικτικό αποτέλεσμα ελέγχων φαίνεται δίπλα. Το **ζητούμενο** είναι να μην υπάρχει η λέξη **FAILED**, να εμφανίζεται μόνο το **Ok** και η επιστρεφόμενη τιμή να είναι **0**.



**TESTING MODE!**

```
Test ARC_NORMAL_VALUE_TESTS :  
  VALUE_1_2 FAILED : arc(1.2) returns unexpected result!  
ARC_NORMAL_VALUE_TESTS FAILED!  
Test aktina_0 : Ok  
Test aktina_PI : Ok  
Process finished with exit code 1
```

# Ερωτήσεις?

---

- Διαβάστε τις σημειώσεις, διαβάστε τις διαφάνειες και δείτε τα videos **πριν** ρωτήσετε
- **Συμβουλευτείτε** τη σελίδα ερωταποκρίσεων του μαθήματος

<https://qna.c-programming.allos.gr>

- **Στείλτε** τις ερωτήσεις σας πριν και μετά το μάθημα στο

[c-programming@allos.gr](mailto:c-programming@allos.gr)

- Εάν έχετε **πρόβλημα** με κάποιο κώδικα στείλτε μαζί τον κώδικα και τα μηνύματα λάθους από το CLI ως κείμενα με copy/paste. Εάν θεωρείτε ότι επιπλέον βοηθά και ένα στιγμιότυπο οθόνης, είναι καλοδεχούμενο.
- Επαναλαμβάνουμε : Μην στείλετε ποτέ κώδικα ως εικόνα μας είναι παντελώς άχρηστος!

