

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ



ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ  
UNIVERSITY OF WEST ATTICA

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ  
ΥΠΟΛΟΓΙΣΤΩΝ

## ΕΡΓΑΣΙΑ 1 ΕΙΣΑΓΩΓΗ

### ΣΤΟΙΧΕΙΑ ΦΟΙΤΗΤΗ:

---

**ΟΝΟΜΑΤΕΠΩΝΥΜΟ:** ΑΘΑΝΑΣΙΟΥ ΒΑΣΙΛΕΙΟΣ ΕΥΑΓΓΕΛΟΣ

**ΑΡΙΘΜΟΣ ΜΗΤΡΩΟΥ:** 19390005

**ΚΑΤΑΣΤΑΣΗ ΦΟΙΤΗΤΗ:** ΠΡΟΠΤΥΧΙΑΚΟ

**ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ:** ΠΑΔΑ

**ΤΜΗΜΑ ΕΡΓΑΣΤΗΡΙΟΥ:** Μ2

**ΚΑΘΗΓΗΤΗΣ ΕΡΓΑΣΤΗΡΙΟΥ:** ΓΕΩΡΓΙΟΣ ΜΕΛΕΤΙΟΥ

**ΗΜΕΡΟΜΗΝΙΑ ΟΛΟΚΛΗΡΩΣΗΣ ΤΗΣ ΕΡΓΑΣΙΑΣ:** 29/10/2021

# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

## ΠΕΡΙΕΧΟΜΕΝΑ

### ΣΕΛΙΔΕΣ 3-5:

#### ΑΠΑΝΤΗΣΕΙΣ ΣΤΑ ΘΕΜΑΤΑ ΘΕΩΡΙΑΣ:

- ΘΕΜΑ 1a (ΣΕΛΙΔΕΣ 3)
- ΘΕΜΑ 1b (ΣΕΛΙΔΕΣ 3-4)
- ΘΕΜΑ 1c (ΣΕΛΙΔΕΣ 4-5)

### ΣΕΛΙΔΕΣ 5-6:

#### ΠΗΓΑΙΟΙ ΚΩΔΙΚΕΣ

- ΠΡΟΓΡΑΜΜΑ «C1a(Wrong)» (ΣΕΛΙΔΕΣ 5-6)
- ΠΡΟΓΡΑΜΜΑ «C1a(Correct)» (ΣΕΛΙΔΕΣ 6)

### ΣΕΛΙΔΕΣ 7-11:

#### ΑΠΑΝΤΗΣΕΙΣ ΣΤΑ ΘΕΜΑΤΑ ΣΧΕΤΙΚΑ ΜΕ ΤΟΥΣ ΠΗΓΑΙΟΥΣ ΚΩΔΙΚΕΣ

- ΘΕΜΑ 2 (ΣΕΛΙΔΕΣ 7-8)
- ΘΕΜΑ 3 (ΣΕΛΙΔΕΣ 9-11)

# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

## ΑΠΑΝΤΗΣΕΙΣ ΣΤΑ ΘΕΜΑΤΑ ΘΕΩΡΙΑΣ

### Θέμα 1

**a) Τι είναι πρόγραμμα:**

Πρόγραμμα είναι μία ακολουθία εντολών που δίνει ο εκάστοτε χρήστης, η οποία αποσκοπεί στην επίλυση ενός προβλήματος που τον απασχολεί.

**b) Ποια είναι τα βήματα ανάπτυξης και εκτέλεσης ενός C προγράμματος:**

Τα βήματα ανάπτυξης και εκτέλεσης ενός C προγράμματος είναι τα εξής:

*i) Σύνταξη (Editing):*

Ο χρήστης έχοντας επίγνωση για τυχόν συντακτικά και λογικά λάθη, συντάσσει το πρόγραμμα του με τις εντολές που γνωρίζει μέσα από το περιβάλλον της γλώσσας C με σκοπό να επιλύσει το πρόβλημα που τον απασχολεί.

*ii) Αποθήκευση του πρόχειρου προγράμματος (Program.c):*

Ο χρήστης, αφού ολοκληρώσει την σύνταξη του προγράμματος του το αποθηκεύει σε «.c» αρχείο (π.χ Program.c), ώστε το πρόγραμμα να λειτουργήσει δίχως περαιτέρω προβλήματα.

*iii) Μεταγλώττιση (Compilation):*

Στη συνέχεια, ο μεταγλωττιστής της γλώσσας προγραμματισμού C (compiler) εντοπίζει τυχόν συντακτικά λάθη στο πρόγραμμα του χρήστη με βάση τους κανόνες δόμησης της γλώσσας, εφόσον και υπάρχουν. Έπειτα, προειδοποιεί τον ίδιο αναλυτικά υπογραμμίζοντας τις λανθασμένες συντάξεις των εντολών με κόκκινο χρώμα και αναφέροντας, μάλιστα, σ' ένα παράθυρο κάτω απ' τον πηγαίο κώδικα, σε ποιες γραμμές εντόπισε τα χαρακτηριστικά λάθη και τι ακριβώς να διορθώσει.

*iv) Μεταγλώττιση του διορθωμένου προγράμματος σε γλώσσα μηχανής (Program.obj):*

Ο χρήστης διορθώνει τα συντακτικά λάθη που εντόπισε ο μεταγλωττιστής (compiler) και του εμφάνισε. Ύστερα, ο μεταγλωττιστής μεταγλωττίζει το πρόγραμμα σε γλώσσα μηχανής (στην γλώσσα που κατανοεί ο επεξεργαστής) και το αποθηκεύει σ' ένα αρχείο «.obj» (π.χ. Program.obj), ώστε ο επεξεργαστής του υπολογιστή να επεξεργαστεί τα δεδομένα του προγράμματος.

# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

## ν) Σύνδεση (Linking):

Ο συνδέτης (linker) συνδέει το αντικειμενικό πρόγραμμα του χρήστη (Program.obj) με άλλα παράγωγα προγράμματα. Τα συγκεκριμένα προγράμματα εκτελούν εντολές στις οποίες τις παίρνει έτοιμες ο χρήστης από τις βιβλιοθήκες που έχει χρησιμοποιήσει στο πρόγραμμά του (π.χ η βιβλιοθήκη «stdio.h» που χρησιμοποιείται για τις εντολές εισόδου/εξόδου δεδομένων, όπως είναι η «printf()» και η «scanf()»). Τέλος, παίρνει το «.obj» αρχείο και παράγει το εκτελέσιμο αρχείο «.exe» (π.χ. Program.exe).

## vi) Εκτελέσιμο αρχείο (Program.exe)

Εμφανίζεται στην οθόνη του χρήστη το εκτελέσιμο αρχείο «.exe», το οποίο ανήκει στο λειτουργικό σύστημα και είναι έτοιμο για καταγραφή και επεξεργασία δεδομένων πάνω στο αντικειμενικό πρόγραμμα (.obj) που έχει αποθηκεύσει ο μεταγλωττιστής.

## vii) Εκτέλεση (Running):

Το πρόγραμμα τρέχει και εκτελούνται όλες οι εντολές που έχει συντάξει ο χρήστης.

## c) Ποιες είναι οι γενικές κατηγορίες λαθών των προγραμμάτων;

Οι γενικές κατηγορίες λαθών των προγραμμάτων είναι οι εξής:

### i) Συντακτικό Λάθος (Syntax Error):

Πρόκειται για λάθος στην σύνταξη του προγράμματος που δεν ανταποκρίνεται στους κανόνες δόμησης της χαρακτηριστικής γλώσσας προγραμματισμού. Για παράδειγμα, στην γλώσσα «C» η απώλεια του τερματικού ελληνικού ερωτηματικού μετά από κάθε εντολή είναι συντακτικό λάθος που εντοπίζει ο μεταγλωττιστής (compiler).

### ii) Λογικό Λάθος (Logic Error):

Είναι λάθος στο οποίο ο μεταγλωττιστής (compiler) δεν εντοπίζει και αναφέρεται στους κανόνες λογικής που ο ίδιος ο χρήστης γνωρίζει. Για παράδειγμα, θέλει να υπολογίσει την πρόσθεση δύο αριθμών ας πούμε «5» και «3» και καταλάθος πληκτρολογεί την εντολή «5 - 3». Ο μεταγλωττιστής δεν θα εντοπίσει κανένα λάθος και θα εκτελέσει την εντολή κανονικά, αλλά δεν θα δώσει το επιθυμητό αποτέλεσμα που επιδιώκει ο χρήστης, καθώς θέλει να υπολογίσει την πρόσθεση των δύο αριθμών ( $5 + 3 = 8$ ) και άθελα θα υπολογίσει την αφαίρεση των δύο αριθμών ( $5 - 3 = 2$ ).

# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

## iii) Λάθος κατά τον χρόνο εκτέλεσης (Runtime Error):

Πρόκειται για το λάθος που εντοπίζεται κατά την εκτέλεση του προγράμματος και δεν είναι εφικτή η διόρθωσή του, καθώς οι εντολές εκτελούνται με βάση το πρόγραμμα που έχει μεταγλωττιστεί και αποθηκευτεί. Συνεπώς, ο χρήστης για να το διορθώσει θα πρέπει να επιστρέψει στο αρχικό βήμα ανάπτυξης και εκτέλεσης ενός προγράμματος (Σύνταξη) και να επαναλάβει την διαδικασία (βλ. Θέμα 1β σελίδες 3-5). Για παράδειγμα, θέλει να υπολογίσει την διαίρεση «x / y» δύο μεταβλητών «x» και «y». Εκχωρεί στην μεταβλητή «y» που είναι ο διαιρέτης την τιμή «0». Το πρόγραμμα θα εκτελεστεί κανονικά αλλά το αποτέλεσμα, προφανώς, δεν θα είναι το επιθυμητό, καθώς γνωρίζουμε από τα μαθηματικά ότι δεν προσδιορίζεται διαίρεση με διαιρέτη το «0».

## ΠΗΓΑΙΟΙ ΚΩΔΙΚΕΣ

### Επισήμανση:

Το πρόγραμμα «C1aWrong» είναι το πρότυπο με τα συντακτικά λάθη, ενώ το πρόγραμμα «C1aCorrect» είναι το συντακτικά διορθωμένο πρόγραμμα.

### ΠΡΟΓΡΑΜΜΑ «C1aWrong»

```
1  #include <stdio.h>
2
3  int main(int argc, int **argv)
4  {
5      int A, B;
6      int C, D, E, F;
7      system ("chcp 1253"); //Διαγραφή αν τρέχει σε Linux
8      printf ("Βασικές αριθμητικές πράξεις με Άκεραίους\n");
9      printf("=====\n\n");
10     printf ("Εισάγετε τον πρώτο αριθμό : ");
11     scanf ("%d",&A);
12     printf ("Εισάγετε το δεύτερο αριθμό : ");
13     scanf ("%d", &B);
14     C = A + B;
15     D = A - B;
```

# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

```
16  E = A * B;
17  F = A / B;
18  printf ("Αθροισμα   : %d\n", c);
19  printf ("Διαφορά    : %d\n", D);
20  printf ("Γινόμενο     : %d\n", E);
21  printf ("Πηλίκο      : %d\n", F);
22  return 0;;
23 }
```

## ΠΡΟΓΡΑΜΜΑ «C1aCorrect»

```
1  #include <stdio.h>
2
3  int main(int argc, int **argv)
4  {
5      int A, B;
6      int C, D, E, F;
7      system ("chcp 1253"); //Διαγραφή αν τρέχει σε Linux
8      printf ("Βασικές αριθμητικές πράξεις με Ακεραίους\n");
9      printf("=====\n\n");
10     printf ("Εισάγετε τον πρώτο αριθμό   : ");
11     scanf ("%d", &A);
12     printf ("Εισάγετε το δεύτερο αριθμό : "); //Απώλεια εισαγωγικού (")
13     //στο τέλος της παραμέτρου της συνάρτησης
14     scanf ("%d", &B);
15     C = A + B;
16     D = A - B;
17     E = A * B;
18     F = A / B;
19
20     printf ("Αθροισμα   : %d\n", C); //Μεταβλητή "c" που δεν έχει δηλωθεί
21     //αντί για "C" που έχει δηλωθεί
22     printf ("Διαφορά    : %d\n", D);
23     printf ("Γινόμενο     : %d\n", E); //Μεταβλητή το ελληνικό "E" που δεν
24     //έχει δηλωθεί αντί για το λατινικό "E" που έχει δηλωθεί
25     printf ("Πηλίκο      : %d\n", F); //Η συνάρτηση "prantf" δεν έχει
26     //δηλωθεί αντί της "printf" που έχει δηλωθεί
27     return 0;;
28 }
```

# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

## ΑΠΑΝΤΗΣΕΙΣ ΣΤΑ ΘΕΜΑΤΑ ΣΧΕΤΙΚΑ ΜΕ ΤΟΥΣ ΠΗΓΑΙΟΥΣ ΚΩΔΙΚΕΣ

### Θέμα 2

Τα συντακτικά λάθη του προγράμματος «C1a(Wrong)» είναι τα εξής:

i) Απώλεια εισαγωγικού (") στο τέλος της παραμέτρου της συνάρτησης (γραμμή 12):

Η παράμετρος της συνάρτησης «printf()» συντάσσεται με διπλά εισαγωγικά ("..."). Στη γραμμή 12 παρατηρείται απώλεια του εισαγωγικού στο κλείσιμο της παραμέτρου.

C1aWrong:

```
12 printf ("Εισάγετε το δεύτερο αριθμό : );
```

C1aCorrect:

```
12 printf ("Εισάγετε το δεύτερο αριθμό : ");
```

ii) Μεταβλητή "c" που δεν έχει δηλωθεί αντί για "C" που έχει δηλωθεί (γραμμή 18):

Στην γραμμή 18 το πρόγραμμα καλεί την συνάρτηση «printf()», ώστε να τυπώσει το περιεχόμενο της μεταβλητής «c». Ο μεταγλωττιστής (compiler) δεν αναγνωρίζει την συγκεκριμένη μεταβλητή γιατί δεν έχει γίνει η τυπική δήλωση της στις πρώτες γραμμές του κώδικα (βλ. γραμμές 5-6). Αντιθέτως, έχει δηλωθεί η μεταβλητή «C» και όπως γνωρίζουμε η γλώσσα «C» είναι «case sensitive» γλώσσα, δηλαδή, οι μεταβλητές «C» και «c» είναι διαφορετικές. Συνεπώς, υπάρχει συντακτικό λάθος και στην θέση της «c» θα βάλουμε την «C».

C1aWrong:

```
18 printf ("Αθροισμα : %d\n", c);
```

C1aCorrect:

```
18 printf ("Αθροισμα : %d\n", C);
```

iii) Μεταβλητή το ελληνικό "Ε" που δεν έχει δηλωθεί αντί για το λατινικό "E" που έχει δηλωθεί (γραμμή 20):

# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

Στη γραμμή 20 καλεί τη συνάρτηση «printf()», ώστε να τυπώσει το περιεχόμενο της μεταβλητής «E». Ο μεταγλωττιστής (compiler) δεν αναγνωρίζει την συγκεκριμένη μεταβλητή γιατί δεν έχει γίνει η τυπική δήλωση της στις πρώτες γραμμές του κώδικα (βλ. Γραμμές 5-6). Μάλιστα, δεν εντοπίζεται με «γυμνό μάτι» το λάθος που παρουσιάζει ο μεταγλωττιστής (compiler). Η σωστή διατύπωση του λάθους είναι πως στο συγκεκριμένο πρόγραμμα η συνάρτηση «printf()» στη γραμμή 20 καλείται να τυπώσει το περιεχόμενο της μεταβλητής «E», όπου ο χαρακτήρας «E» είναι στην ελληνική έκδοση, ενώ στην γραμμή 6 έχει δηλωθεί η αντίστοιχη μεταβλητή «Ε», όπου ο χαρακτήρας «Ε» είναι στην λατινική έκδοση. Συνεπώς, υπάρχει συντακτικό λάθος και στην γραμμή 20 θα κάνουμε την χαρακτηριστική αλλαγή.

C1aWrong:

```
20 printf ("Γινόμενο : %d\n", E);
```

C1aCorrect:

```
20 printf ("Γινόμενο : %d\n", E);
```

*iv) Η συνάρτηση "prantf" δεν έχει δηλωθεί αντί της "printf" που έχει δηλωθεί (γραμμή 21):*

Στη γραμμή 21 το πρόγραμμα καλεί την συνάρτηση «prantf()». Ο μεταγλωττιστής (compiler) δεν εντοπίζει συντακτικό λάθος, αλλά λάθος βρίσκει ο συνδέτης (linker). Συγκεκριμένα, δεν αναγνωρίζει την συνάρτηση αυτή από καμία βιβλιοθήκη που έχει εισαχθεί στο πρόγραμμα. Ο συντάκτης του προγράμματος, πιο αναλυτικά, έχει εισάγει την βιβλιοθήκη «stdio.h» που παίρνει έτοιμες τις συναρτήσεις εισόδου/εξόδου δεδομένων «printf()» και «scanf()». Συνεπώς, πληκτρολόγησε λανθασμένα την συνάρτηση που ήθελε να βάλει, καθώς ο σκοπός του θα ήταν να τυπώσει το περιεχόμενο μίας μεταβλητής (συγκεκριμένα της «F») και γι' αυτό το λόγο διορθώνουμε το λάθος αντικαθιστώντας την «prantf()» με την «printf()» που οδηγεί στο επιθυμητό αποτέλεσμα του συντάκτη.

C1aWrong:

```
21 prantf ("Πηλίκο : %d\n", F);
```

C1aCorrect:

```
21 printf ("Πηλίκο : %d\n", F);
```



# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

## Θέμα 3

### Επισημάνση:

Παρακάτω παρουσιάζονται τρία χαρακτηριστικά παραδείγματα (Παράδειγμα 1, Παράδειγμα 2, Παράδειγμα 3) πάνω στο πρόγραμμα «C1aCorrect» εκχωρώντας διαφορετικές τιμές στις δύο μεταβλητές.

### Παράδειγμα 1:

Active code page: 1253

Βασικές αριθμητικές πράξεις με Ακεραίους

=====

Εισάγετε τον πρώτο αριθμό : 10

Εισάγετε το δεύτερο αριθμό : 5

Άθροισμα : 15

Διαφορά : 5

Γινόμενο : 50

Πηλίκο : 2

### Παράδειγμα 2:

Active code page: 1253

Βασικές αριθμητικές πράξεις με Ακεραίους

=====

Εισάγετε τον πρώτο αριθμό : 5

Εισάγετε το δεύτερο αριθμό : 10

Άθροισμα : 15

Διαφορά : -5

Γινόμενο : 50

Πηλίκο : 0

# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

Active code page: 1253

Βασικές αριθμητικές πράξεις με Ακεραίους

=====

Εισάγετε τον πρώτο αριθμό : 54645234234

Εισάγετε το δεύτερο αριθμό : 35453453454

Άθροισμα : -95625528

Διαφορά : 2011911596

Γινόμενο : -695536596

Πηλίκο : -1

## Παρατηρήσεις:

*i) Παράδειγμα 1 (πρώτος αριθμός = 10, δεύτερος αριθμός = 5):*

Δεν παρατηρείται καμία ενδιαφέρουσα παρατήρηση, καθώς τα αποτελέσματα στον προγραμματισμό και στα μαθηματικά είναι αναμενόμενα. Πιο αναλυτικά, η πρόσθεση είναι « $10 + 5 = 15$ », η αφαίρεση είναι « $10 - 5 = 5$ », ο πολλαπλασιασμός είναι « $10 * 5 = 50$ » και η διαίρεση είναι « $10 / 5 = 2$ ».

*ii) Παράδειγμα 2 (πρώτος αριθμός = 5, δεύτερος αριθμός = 10):*

Εδώ παρατηρείται μία ενδιαφέρουσα παρατήρηση που προγραμματιστικά είναι αναμενόμενη αλλά όχι το ίδιο μαθηματικά. Πιο αναλυτικά, η πρόσθεση είναι « $5 + 10 = 15$ », η αφαίρεση είναι « $5 - 10 = -5$ », ο πολλαπλασιασμός είναι « $5 * 10 = 50$ » και η διαίρεση είναι « $5 / 10 = 0$ ». Μαθηματικά και αναλυτικά η διαίρεση « $5 / 10$ » έχει πηλίκο «0.5» και υπόλοιπο «5». Στο πρόγραμμα το πηλίκο είναι «0», επειδή ο χρήστης έχει δηλώσει την μεταβλητή που θα εκχωρήσει το πηλίκο ως ακέραια (int).

# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

iii) Παράδειγμα 3 (πρώτος αριθμός = 54645234234, δεύτερος αριθμός = 35453453454)

Εδώ παρατηρούνται ενδιαφέρουσες παρατηρήσεις, καθώς τα αποτελέσματα είναι μη αναμενόμενα και προφανώς μη επιθυμητά. Πιο αναλυτικά, η πρόσθεση είναι « $54645234234 + 35453453454 = -95625528$ », η αφαίρεση είναι « $54645234234 - 35453453454 = 2011911596$ », ο πολλαπλασιασμός είναι « $54645234234 * 35453453454 = -695536596$ » και η διαίρεση είναι « $54645234234 / 35453453454 = -1$ ». Η εξήγηση πίσω από αυτά τα περιέργα αποτελέσματα είναι η εξής: το εύρος τιμών σε «bit» στη μνήμη ενός «32-bit» επεξεργαστή είναι 0-255. Συνεπώς, τα νούμερα αυτά είναι εκτός του εύρους τιμών, με αποτέλεσμα να παρουσιάζονται προβλήματα στην καταχώρηση των δεδομένων στα κελιά μνήμης και εξού αυτές οι ανακρίβειες στα αποτελέσματα.

# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ



Σας ευχαριστώ για την προσοχή σας.

