Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Department of Electrical and Computer Engineering

ΔΟΜΗΜΕΝΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

ΕΡΓΑΣΤΗΡΙΟ 9

Διδάσκουσα καθηγήτρια: **Ε.Κατσίρη**

Υπεύθυνος εργαστηρίου: **Α.Γαζής**

Βοηθός εργαστηρίου: **Χ.Καρασούλας**

ΘΕΩΡΙΑ

Όταν δηλώνουμε μια μεταβλητή στον compiler, (π.χ. int x=3;), αυτή η δήλωση δεσμεύει χώρο στην μνήμη του υπολογιστή!

Ως «μνήμη υπολογιστή», ορίζεται η μνήμη RAM

Οι δείκτες έχουν συγκεκριμένο μέγεθος ανάλογα με την αρχιτεκτονική του συστήματος:

- **→Αρχιτεκτονική 32 bit →μέγεθος δείκτη: 4 bytes (32 bits)**
- **→Αρχιτεκτονική 64 bit →μέγεθος δείκτη: 8 bytes (64 bits)**
- **→Αρχιτεκτονική 16 bit →μέγεθος δείκτη: 2 bytes (16 bits) (παλαιότερα υπολογιστικά συστήματα)**

- **Οι δείκτες στην C καταλαμβάνουν (δεσμεύουν) κελιά μνήμης όπου κάθε κελί μνήμης έχει χωρητικότητα 1Byte!**
- **Κάθε κελί είναι αριθμημένο με έναν ΜΟΝΑΔΙΚΟ ακέραιο αριθμό (το κάνει ο DevCpp δλδ ο compiler)**

ΘΕΩΡΙΑ

Οι <u>δείκτες</u> στην C καταλαμβάνουν κελιά μνήμης όπου κάθε κελί μνήμης έχει χωρητικότητα 1Byte!

Η μνήμη έχει την εξής αναπαράσταση:

0	1	2	3	….	1000	….
1 byte	1 byte	1 byte	1 byte	1 byte	1 byte	1 byte

ή

0	1 byte
1	1 byte
3	1 byte
….	
1000	1 byte
….	1 byte

ΘΕΩΡΙΑ

Η δήλωση μιας μεταβλητής δεσμεύει διαδοχικές θέσεις μνήμης (ανάλογα με το τύπο δεδομένων)

```
int gazis

//int:integer-> 4
      bytes
```

...	150	151	152	153	154	155	156	157	158	159	...
...	////	////	////	////							...

Όπου //// αναπαραστά δεσμευμένο κελί μνήμης

Πως βλέπω-ξέρω την χωρητικότητα εντός τύπου μεταβλητής (int, float,char κτλ)?

- **Για να ελέγξουμε την χωρητικότητα κάθε τύπου μεταβλητής:**

```
printf("%d", sizeof(int)); printf("%d", sizeof(float)); …
```

ΘΕΩΡΙΑ

Πως βλέπω-ξέρω την χωρητικότητα εντός τύπου μεταβλητής (int, float,char κτλ)?

```c
#include <stdio.h>

main(){
 printf("sizeof(char)        == %d\n", sizeof(char));
 printf("sizeof(short)       == %d\n", sizeof(short));
 printf("sizeof(int)         == %d\n", sizeof(int));
 printf("sizeof(long)        == %d\n", sizeof(long));
 printf("sizeof(float)       == %d\n", sizeof(float));
 printf("sizeof(double)      == %d\n", sizeof(double));
 printf("sizeof(long double) == %d\n", sizeof(long double));
 printf("sizeof(long long)   == %d\n", sizeof(long long));
}
```

Αποτελέσματα κώδικα

```
sizeof(char)        == 1
sizeof(short)       == 2
sizeof(int)         == 4
sizeof(long)        == 4
sizeof(float)       == 4
sizeof(double)      == 8
sizeof(long double) == 16
sizeof(long long)   == 8
```

Παράδειγμα 1

Να κατασκευάσετε ένα πρόγραμμα το οποίο:

1. **Θα ορίζει μια ακέραια τιμή σε μια μεταβλητή (int) και θα την αρχικοποιεί με μια τυχαία τιμή της αρεσκείας σας**
2. **Θα ορίζει μια μη ακέραια τιμή σε μια μεταβλητή (float) και θα την αρχικοποιεί με μια τυχαία τιμή της αρεσκείας σας**
3. **Θα ορίζει μια αλφαριθμητική τιμή σε μια μεταβλητή (char) και θα την αρχικοποιεί με μια τυχαία τιμή της αρεσκείας σας**
4. **Θα εκτυπώνει την τιμή των παραπάνω μεταβλητών**
5. **Θα ορίζει και θα εκτυπώνει το μέγεθος των παραπάνω μεταβλητών**
6. **Θα ορίζει και θα εκτυπώνει τη διεύθυνση μνήμης των παραπάνω μεταβλητών(pointer)**

- **Το πρόγραμμα απαιτείται να μεταγλωττίζεται και να εκτελείται επιτυχώς .**

Παράδειγμα 2

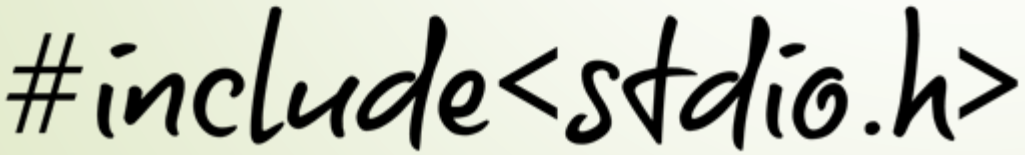
Να κατασκευάσετε ένα πρόγραμμα το οποίο:

1. **Θα δέχεται από τον χρήστη μια *FLOAT* τιμή, θα την αποθηκεύει σε μια μεταβλητή και θα εκτυπώνει το κελί μνήμης(address) που έχει αποθηκευτεί.**
2. **Θα έχει αποθηκευμένους 2 πίνακες με τις εξής τιμές:**
   A. **{-31,-12,21,330,11,0,-43}**
   B. **{-9,1.94,32.5,1.33,11.4}**

   και θα εκτυπώνει για το κάθε στοιχείο του πίνακα Α και πίνακα Β
3. **Θα εκτυπώνει για το 1° στοιχείο του πίνακα Α και το 3° του πίνακα Β την τιμή και την διεύθυνσή του (κελί μνήμης-pointer)**

- **Το πρόγραμμα απαιτείται να μεταγλωττίζεται και να εκτελείται επιτυχώς .**

Ευχαριστούμε για την προσοχή σας



- Για απορίες ή διευκρινίσεις, παρακαλώ όπως επικοινωνήστε μαζί μας στα κάτωθι email:

ekatsiri@ee.duth.gr
agazis@ee.duth.gr
ckarasou@ee.duth.gr