

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ & ΠΛΗΡΟΦΟΡΙΚΗΣ



ΕΙΣΑΓΩΓΗ ΣΤΟ ΔΙΑΔΙΚΑΣΤΙΚΟ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ (2009-2010)
ΥΠΕΥΘΥΝΟΙ ΔΙΔΑΣΚΟΝΤΕΣ ΕΡΓΑΣΤΗΡΙΟΥ: Α. ΦΩΚΑ, Κ. ΣΤΑΜΟΣ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ & ΠΛΗΡΟΦΟΡΙΚΗΣ

4^ο ΣΕΤ ΑΣΚΗΣΕΩΝ

Οι ασκήσεις αυτού του φυλλαδίου καλύπτουν τα παρακάτω θέματα και δίνονται ενδεικτικά οι αντίστοιχες ενότητες στο βιβλίο The GNU C Programming Tutorial που μπορείτε συμβουλευτείτε (<http://crasseux.com/books/ctutorial/>) .

- Δείκτες (κεφάλαιο Pointers)
- Δομές (κεφάλαιο Data Structures)

Άσκηση 1η

Γράψτε μία συνάρτηση *strmix(s,t)* η οποία παίρνει ως παράμετρο τα αλφαριθμητικά *s* και *t* και επιστρέφει ένα νέο αλφαριθμητικό το οποίο περιέχει εναλλάξ τους χαρακτήρες των δύο αλφαριθμητικών. Η συνάρτηση θα πρέπει να δουλεύει με αλφαριθμητικά οποιουδήποτε μήκους (επομένως χρησιμοποιήστε δείκτες). Το κυρίως πρόγραμμα θα ζητάει από το χρήστη δύο αλφαριθμητικά μεγέθους το πολύ N (το N θα το δίνει επίσης ο χρήστης) και θα τυπώνει το αποτέλεσμα της *strmix*.

Π.χ. (με έντονα η είσοδος του χρήστη):

```
Dwse megisto mege8os string:40
Dwse prwto string (mege8os to poly 40):1234567890
Dwse deytero string (mege8os to poly 40):abcdefg
1a2b3c4d5e6f7g890
```

Άσκηση 2η

Γράψτε μία συνάρτηση *replace(input, initial, replacement)* η οποία παίρνει ως παράμετρο τα αλφαριθμητικά *input* , *initial* και *replacement* και επιστρέφει ένα νέο αλφαριθμητικό στο οποίο όλες οι εμφανίσεις του *initial* στο *input* έχουν αντικατασταθεί με το *replacement*. Η συνάρτηση θα πρέπει να δουλεύει με αλφαριθμητικά οποιουδήποτε μήκους (επομένως χρησιμοποιήστε δείκτες). Το κυρίως πρόγραμμα θα ζητάει από το χρήστη τρία αλφαριθμητικά μεγέθους το πολύ N (το N θα το δίνει επίσης ο χρήστης) και θα τυπώνει το αποτέλεσμα της *replace*.

Π.χ. (με έντονα η είσοδος του χρήστη):

```
Dwse megisto mege8os string:30
Dwse string eisodou (mege8os to poly 30):123456789012345678901234
Dwse string pros antikatastasi(mege8os to poly 30):456
Dwse neo string (mege8os to poly 30):TesseraPenteEksi
123TesseraPenteEksi7890123TesseraPenteEksi78901234
```

Άσκηση 3η

Να υλοποιήσετε πρόγραμμα με χρήση δεικτών το οποίο θα επιτρέπει στο χρήστη να επιλέγει μία από τις παρακάτω λογικές πράξεις. Στη συνέχεια ανάλογα με την επιλογή του χρήστη θα ζητά είσοδο η οποία θα δίνεται ως δυαδικός αριθμός. Τέλος θα εφαρμόζει στη δυαδική ακολουθία την επιλεγμένη πράξη και θα εκτυπώνει το αποτέλεσμα.

- a. XOR
- b. OR
- c. AND
- d. COMPLEMENT
- e Shift left
- f. Shift right

Προσοχή, το πρόγραμμά σας δεν θα κάνει χρήση των bitwise operators της C, αλλά θα εφαρμόζει την επιλεγμένη λογική πράξη σε κάθε ψηφίο του δυαδικού αριθμού.

Λεπτομέρειες για τις λογικές πράξεις στο http://en.wikipedia.org/wiki/Bitwise_operation

```

1. XOR
2. OR
3. AND
4. COMPLEMENT
5. SHIFT LEFT
6. SHIFT RIGHT
0. exodos
Dwse epilogh: 1
Dwse 1o dyadiko (mexri 16 bit): 0101
Dwse 2o dyadiko (mexri 16 bit): 0011
Apotelesma XOR: 0110
1. XOR
2. OR
3. AND
4. COMPLEMENT
5. SHIFT LEFT
6. SHIFT RIGHT
0. exodos
Dwse epilogh: 2
Dwse 1o dyadiko (mexri 16 bit): 0101
Dwse 2o dyadiko (mexri 16 bit): 0011
Apotelesma OR: 0111
1. XOR
2. OR
3. AND
4. COMPLEMENT
5. SHIFT LEFT
6. SHIFT RIGHT
0. exodos
Dwse epilogh: 3
Dwse 1o dyadiko (mexri 16 bit): 0101
Dwse 2o dyadiko (mexri 16 bit): 0011
Apotelesma AND: 0001
1. XOR
2. OR
3. AND

```



```

4. COMPLEMENT
5. SHIFT LEFT
6. SHIFT RIGHT
0. exodos
Dwse epilogh: 4
Dwse dyadiko (mexri 16 bit): 0101
Apotelesma COMPLEMENT: 1010
1. XOR
2. OR
3. AND
4. COMPLEMENT
5. SHIFT LEFT
6. SHIFT RIGHT
0. exodos
Dwse epilogh: 5
Dwse dyadiko (mexri 16 bit): 00010111
Dwse 8eseis shift: 1
Apotelesma SHIFT LEFT: 00101110
1. XOR
2. OR
3. AND
4. COMPLEMENT
5. SHIFT LEFT
6. SHIFT RIGHT
0. exodos
Dwse epilogh: 6
Dwse dyadiko (mexri 16 bit): 00010111
Dwse 8eseis shift: 1
Apotelesma SHIFT RIGHT: 00001011

```

Άσκηση 4^η

Σας ζητείτε να αποθηκεύσετε σε έναν πίνακα μεγέθους N στοιχεία τύπου struct employee. Κάθε καινούριο στοιχείο αποθηκεύεται στην τελευταία άδεια θέση του πίνακα και μπορεί να διαγραφεί μόνο το τελευταίο στοιχείο που έχει εισαχθεί στον πίνακα. Δίνεται:

```

struct employee
{
    char *firstName;
    char *lastName;
    int age;
};

```

Υλοποιήστε τις παρακάτω συναρτήσεις:

```

/* δημιουργία του πίνακα */
/* pos : πρώτη ελεύθερη θέση στον πίνακα */
/* επιστρέφει πίνακα με N στοιχεία τύπου struct employee */
struct employee* create( int *pos);

/* εισαγωγή στοιχείου item στον πίνακα pin */
/* pos : πρώτη ελεύθερη θέση στον πίνακα */
void ins(struct employee* pin, int *pos, struct employee* item);

/* διαγραφή στοιχείου από τον πίνακα */

```

```

/* pos : πρώτη ελεύθερη θέση στη στοίβα */
struct employee* del(struct employee* pin, int *pos);

/*εκτύπωση πίνακα */
/* pos : πρώτη ελεύθερη θέση στη στοίβα */
void printpin(struct employee* pin, int *pos);

/* ταξινόμηση πίνακα */
/* pos : πρώτη ελεύθερη θέση στη στοίβα */
/* field : το πεδίο με βάση το οποίο θέλουμε να γίνει η ταξινόμηση */
/* direction : αν η ταξινόμηση θα είναι αύξουσα τότε (direction=1) */
/* Αν θα είναι φθίνουσα (direction=-1) αντίστοιχα. */
void sort(struct employee* pin, int *pos, char *field, int direction);

```

Γράψτε την main() η οποία εμφανίζει μενού επιλογών στο χρήστη:

1. Εισαγωγή στοιχείων υπαλλήλου
2. Διαγραφή στοιχείων υπαλλήλου
3. Εμφάνιση όλων των υπαλλήλων
4. Ταξινόμηση Πίνακα
5. Έξοδος

Ανάλογα με την επιλογή του χρήστη ζητούνται τα απαραίτητα στοιχεία (π.χ. για την Εισαγωγή Στοιχείων ζητούνται από τον χρήστη το όνομα, επώνυμο και ηλικία) και καλείται η κατάλληλη συνάρτηση.

Ari8mos stoiceiwn ston pinaka 0

```

1. Eisagwgh Stoiceiwn Ypallhlou
2. Diagrafh Stoiceiwn Ypallhlou
3. Emfanish olwn twn Ypallhlwn
4. Taxinomisi
5. Exit
Dwse epilogh: 1

```

Dwse Stoiceia Ypallhlou:

Onoma: **Maria**
 Epwnymo: **Papa**
 Hlikia: **22**

Ari8mos stoiceiwn ston pinaka 1

```

1. Eisagwgh Stoiceiwn Ypallhlou
2. Diagrafh Stoiceiwn Ypallhlou
3. Emfanish olwn twn Ypallhlwn
4. Taxinomisi
5. Exit
Dwse epilogh: 1

```

Dwse Stoixeia Ypallhlou:

Onoma: **Takis**
 Epwnymo: **Spyrou**
 Hlikia: **19**

Ari8mos stoxeiwn ston pinaka 2

1. Eisagwgh Stoxeiwn Ypallhlou
2. Diagrafh Stoxeiwn Ypallhlou
3. Emfanish olwn twn Ypallhlwn
4. Taxinomisi
5. Exit

Dwse epilogh: **3**

Emfanish olwn twn ypallhlwn:

1. Maria Papa 22
2. Takis Spyrou 19

Ari8mos stoxeiwn ston pinaka 2

1. Eisagwgh Stoxeiwn Ypallhlou
2. Diagrafh Stoxeiwn Ypallhlou
3. Emfanish olwn twn Ypallhlwn
4. Taxinomisi
5. Exit

Dwse epilogh: **4**

1. Taxinomisi me onoma
2. Taxinomisi me eponimo
3. Taxinomisi me hlikia

Dwse epilogh: **1**

1. Taxinomisi Auxousa
- 1. Taxinomisi Fthinousa

Dwse epilogh: **1**

1. Maria Papa 22
2. Takis Spyrou 19

Ari8mos stoxeiwn ston pinaka 2

1. Eisagwgh Stoxeiwn Ypallhlou
2. Diagrafh Stoxeiwn Ypallhlou
3. Emfanish olwn twn Ypallhlwn
4. Taxinomisi
5. Exit

Dwse epilogh: **4**

1. Taxinomisi me onoma
2. Taxinomisi me eponimo
3. Taxinomisi me hlikia

Dwse epilogh:

1

1. Taxinomisi Auxousa
- 1. Taxinomisi Fthinousa

Dwse epilogh: **-1**

1. Takis Spyrou 19
2. Maria Papa 22

Ari8mos stoxeiwn ston pinaka 2

1. Eisagwgh Stoxeiwn Ypallhlou

2. Diagrafh Stoixeiwn Ypallhlou
3. Emfanish olwn twn Ypallhlwn
4. Taxinomisi
5. Exit

Dwse epilogh: **4**

1. Taxinomisi me onoma
2. Taxinomisi me eponimo
3. Taxinomisi me hlikia

Dwse epilogh: **3**

1. Taxinomisi Auxousa
- 1. Taxinomisi Fthinousa

Dwse epilogh: **-1**

1. Maria Papa 22
2. Takis Spyrou 19

Ari8mos stoixeiwn ston pinaka 2

1. Eisagwgh Stoixeiwn Ypallhlou
2. Diagrafh Stoixeiwn Ypallhlou
3. Emfanish olwn twn Ypallhlwn
4. Taxinomisi
5. Exit

Dwse epilogh: **2**

Diagrafh Teleutaiou sth Lista Ypallhlou:

Ari8mos stoixeiwn ston pinaka 1

1. Eisagwgh Stoixeiwn Ypallhlou
2. Diagrafh Stoixeiwn Ypallhlou
3. Emfanish olwn twn Ypallhlwn
4. Taxinomisi
5. Exit

Dwse epilogh: **3**

Emfanish olwn twn ypallhlwn:

1. Maria Papa 22

Άσκηση 5η

Δημιουργείστε κώδικα ο οποίος υλοποιεί το ακόλουθο παιχνίδι: αρχικά ζητείται από τον πρώτο χρήστη να εισαγάγει μια λέξη. Στη συνέχεια η οθόνη καθαρίζεται και εμφανίζονται χαρακτήρες '_' στη θέση των γραμμάτων της λέξης καθώς και ένας μετρητής που δηλώνει πόσες προσπάθειες απομένουν στο δεύτερο χρήστη, που καλείται να μαντέψει τη λέξη δίνοντας έναν χαρακτήρα κάθε φορά. Αν ο εισαγόμενος χαρακτήρας υπάρχει εμφανίζεται στην κατάλληλη θέση της λέξης, αν δεν υπάρχει ο μετρητής μειώνεται κατά 1 και αν έχει επιλεχθεί ήδη εμφανίζεται ένα ενημερωτικό μήνυμα. Το παιχνίδι τελειώνει όταν βρεθούν όλα τα γράμματα της λέξης ή όταν μηδενιστεί ο μετρητής (οπότε εμφανίζονται τα κατάλληλα μηνύματα στην οθόνη).

Προτείνεται:

1. να χρησιμοποιηθεί κώδικας για τη μετατροπή πεζών σε κεφαλαία σε μορφή συνάρτησης έτσι ώστε όλοι οι εισαγόμενοι χαρακτήρες να μετατρέπονται σε κεφαλαία

2. ο έλεγχος ύπαρξης ενός χαρακτήρα σε μια λέξη να γίνεται με την κλήση συνάρτησης η οποία θα λαμβάνει ως ορίσματα τη λέξη και τον χαρακτήρα προς αναζήτηση και θα επιστρέφει 0 αν δεν υπάρχει ή 1 αν υπάρχει.

Υπόδειξη: Ο καθαρισμός της οθόνης γίνεται με την εντολή system("clear") (unix) ή system("cls") (DOS)

Xrhsth 1::Dwse thn Krymmenh Leksh (mexri 256 grammata): **hello**

```
-----                Points Left 10
Xrhsth 2::Proteine kapoio gramma:
k
Den bre8hkan 'k' sth leksh.
-----                Points Left 9
Xrhsth 2::Proteine kapoio gramma:
f
Den bre8hkan 'f' sth leksh.
-----                Points Left 8
Xrhsth 2::Proteine kapoio gramma:
l
Bre8hkan 2 'l' sth leksh.
--ll-                Points Left 8
Xrhsth 2::Proteine kapoio gramma:
l
Den bre8hkan 'l' sth leksh.
--ll-                Points Left 7
Xrhsth 2::Proteine kapoio gramma:
h
Bre8hkan 1 'h' sth leksh.
h-ll-                Points Left 7
Xrhsth 2::Proteine kapoio gramma:
e
Bre8hkan 1 'e' sth leksh.
hell-                Points Left 7
Xrhsth 2::Proteine kapoio gramma:
o
Bre8hkan 1 'o' sth leksh.
Bravo! H krymmenh leksh htan: "hello"
```

