

**ΕΡΓΑΣΤΗΡΙΟ 3****Μέρος Α': Πίνακες**

1. Γράψτε ένα πρόγραμμα *mergeTables.c* το οποίο συγχωνεύει δύο ταξινομημένους πίνακες σε ένα τρίτο ταξινομημένο πίνακα. Η ταξινόμηση του τρίτου πίνακα να γίνει καθώς εισάγονται τα στοιχεία σε αυτόν από τους άλλους δυο πίνακες.

**Σημείωση:** Απαγορεύεται η χρήση αλγόριθμου ταξινόμησης

2. Γράψτε ένα πρόγραμμα *roman2arab.c* το οποίο μετατρέπει ένα αριθμό γραμμένο σε ρωμαϊκή μορφή σε αριθμό με αραβικά ψηφία.

Υπόδειξη: Μπορείτε να βρείτε χρήσιμους τους πιο κάτω (παράλληλους) πίνακες

```
char roman[] = "IVXLCDM";
```

```
int arab[] = {1, 5, 10, 50, 100, 500, 1000};
```

3. Γράψτε ένα πρόγραμμα *dutchFlag.c* το οποίο ταξινομεί ένα πίνακα ακεραίων του οποίου τα στοιχεία έχουν μόνο τις τιμές 0, 1 και 2.
4. Γράψτε ένα πρόγραμμα το οποίο βρίσκει όλα τα στοιχεία ενός πίνακα δυο διαστάσεων τα οποία είναι ταυτόχρονα το μέγιστο στοιχείο της γραμμής και μικρότερο στοιχείο της στήλης όπου βρίσκονται.

**Μέρος Β': Δείκτες**

5. Μελετήστε το παρακάτω τμήμα κώδικα, και γράψτε τι θα τυπωθεί από τις εντολές προτού τις εκτελέσετε:

```
int A[] = {12, 23, 34, 45, 56, 67, 78, 89, 90};  
int * p = A;
```

```
printf(" *p+2      = %d\n", *p+2);  
printf(" *(p+2)    = %d\n", *(p+2));  
printf("&p+1       = %p\n", &p+1);  
printf("p+1        = %p\n", p+1);  
printf("&A[4]-3     = %p\n", &A[4]-3);  
printf("A+3         = %p\n", A+3);  
printf("&A[7]-p      = %p\n", &A[7]-p);  
printf("p+( *p-10)  = %p\n", p+( *p-10));
```

6. Συμπληρώστε το δοθέν αρχείο *pointers.c* στο οποίο θα δηλώνεται ένας πίνακας ακεραίων μεγέθους `MAX_SIZE`. Ο χρήστης θα δίνει αρχικά το πλήθος των ακεραίων αριθμών που θέλει να εισάγει. Στη συνέχεια θα εισάγει τους ακεραίους οι οποίοι θα αποθηκεύονται στον πίνακα. Έπειτα θα εκτυπώνεται ο πίνακας και θα καλείται η συνάρτηση `counting(...)` που θα υπολογίζει και θα επιστρέφει το πλήθος των στοιχείων του πίνακα που είναι θετικά,



αρνητικά και ίσα με το μηδέν. Η συνάρτηση πρέπει να επιστρέφει τις 3 αυτές τιμές στη συνάρτηση main. Η συνάρτηση counting πρέπει να είναι τύπου *void*.

Παράδειγμα εκτέλεσης:

```
Dose to megethos toy pinaka (metaksi 1 kai 10): 10
Tha dwseis 10 akeraious arithmous, ena-ena tin fora
Dwse arithmo 1: 1
Dwse arithmo 2: 2
Dwse arithmo 3: 0
Dwse arithmo 4: 12
Dwse arithmo 5: -23
Dwse arithmo 6: 11
Dwse arithmo 7: 0
Dwse arithmo 8: -2
Dwse arithmo 9: 90
Dwse arithmo 10: -23
```

Ektyposi pinaka:

```
table[0] = 1
table[1] = 2
table[2] = 0
table[3] = 12
table[4] = -23
table[5] = 11
table[6] = 0
table[7] = -2
table[8] = 90
table[9] = -23
```

Ο pinakas periexei 5 thetikous(ο) arithmous, 3 arnhtikous arithmous(ο), kai 2 arithmous(ο) iso me mhden.

7. Γράψτε ένα πρόγραμμα το οποίο αντιστρέφει τα περιεχόμενα ενός πίνακα ακεραίων αριθμών (δηλώστε το μέγεθος του πίνακα σαν σταθερά). Όλες οι λειτουργίες του προγράμματος (ανάγνωση αριθμών, αντιστροφή αριθμών, εκτύπωση αριθμών) να υλοποιηθούν μέσω ξεχωριστών συναρτήσεων χρησιμοποιώντας αριθμητική των δεικτών (να μην χρησιμοποιηθούν πουθενά [ ] εκτός στο σημείο δήλωσης του πίνακα).