

ΑΣΚΗΣΕΙΣ-ΑΡΧΕΙΑ

```

    } else if (a) {
        for (; o > i; i++)
            if (r = t.call(e[i], i, e[i]), r !== !1) break
    } else
        for (i in e)
            if (r = t.call(e[i], i, e[i]), r !== !1) break;
    return e
},
trim: b && !b.call("\uffff\u00a0") ? function(e) {
    return null == e ? "" : b.call(e)
} : function(e) {
    return null == e ? "" : (e + "").replace(C, "")
},
makeArray: function(e, t) {
    var n = t || [];
    return null != e && (M(Object(e)) ? x.merge(n, "string" == typeof e ? [e] : e) : h.call(n, e)), n
},
isArray: function(e, t, n) {
    var r;
    if (t) {
        if (n) return m.call(t, e, n);
        for (r = t.length, o = 0; o < r; o++)
            if (h.call(t, o, e)) return !0;
        return !1;
    }
    return m.call(t, e, n);
}

```

1. Να κατασκευαστεί πρόγραμμα που θα ανοίγει ένα αρχείο που περιέχει τα ονόματα των εργαζομένων και τον μισθό τους. Το πρόγραμμα θα εισάγει τα δεδομένα των αρχείων σε πίνακες και θα εξάγει τα δεδομένα των πινάκων σε αρχείο ταξινομημένα με βάση το όνομα του εργαζομένου και θα εμφανίζει τον εργαζόμενο με τον μεγαλύτερο μισθό. Επίσης να κατασκευαστεί συνάρτηση η οποία θα βρίσκει και θα επιστρέφει τον υπάλληλο με τον μεγαλύτερο μισθό.

2. Να γραφεί πρόγραμμα που θα περιέχει τα ακόλουθα:

- Συνάρτηση `readcontent()` η οποία θα ζητάει από τον χρήστη να εισάγει ένα αριθμό τον οποίο και θα επιστρέφει.
- Συνάρτηση `store_to_file(char *name, int prime)`, η οποία θα εισάγει σε ένα αρχείο με όνομα `name` τον αριθμό που δέχεται σαν όρισμα (`prime`).
- Συνάρτηση `sum_prime(int x)`, η οποία για τον αριθμό που δέχεται ως όρισμα θα πραγματοποιεί εύρεση του αθροίσματος όλων των πρώτων αριθμών μέχρι και αυτόν (Πρώτος αριθμός, ένας αριθμός που διαιρείται με τον εαυτό του και την μονάδα).

- Συνάρτηση `display_data(char *fn)`, η οποία θα εμφανίζει τα δεδομένα από ένα αρχείο με όνομα `fn`.
- Κύρια συνάρτηση η οποία θα ζητάει από τον χρήστη να εισάγει μία επιλογή (ακέραιος αριθμός). Αν η επιλογή είναι 1 θα ζητάει από τον χρήστη να εισάγει ένα ακέραιο αριθμό (χρήση συνάρτησης `readcontent`) και θα πραγματοποιεί εύρεση του αθροίσματος των πρώτων αριθμών που περιέχονται σε αυτόν και εκτύπωση του σε αρχείο. Αν η επιλογή είναι 2 θα εμφανίζει τα δεδομένα από το αρχείο και αν η επιλογή είναι τρία θα σταματάει την εκτέλεση του προγράμματος.

Χρήση παρακάτω κώδικα: https://github.com/vasnastos/PROGRAMMING-TO-C-2/blob/master/Course4/%CE%91%CE%A1%CE%A7%CE%95%CE%99%CE%91_EXTRA_%CE%9C%CE%91%CE%98%CE%97%CE%9C%CE%91/%CE%91%CE%A3%CE%9A%CE%97%CE%A3%CE%97_2/primeprogram.c

3. Να γραφεί πρόγραμμα, το οποίο θα ενημερώνει θα μετατρέπει σε ένα προϋπάρχον αρχείο κειμένου όλα τα γράμματα του αγγλικού αλφαβήτου σε κεφαλαία.

4. Να γραφεί πρόγραμμα το οποίο θα περιέχει τα ακόλουθα:

- Συνάρτηση `sort()`, η οποία θα ταξινομεί τα δεδομένα του αρχείου με βάση την τιμή τους.

- Συνάρτηση search, η οποία θα ανοίγει το αρχείο και θα πραγματοποιεί, αναζήτηση ενός προϊόντος από το αρχείο με βάση το όνομα του και αν υπάρχει θα εμφανίζει τις πληροφορίες του αλλιώς να εμφανίζει μήνυμα λάθους.
- Συνάρτηση insert, η οποία θα πραγματοποιεί εισαγωγή ενός προϊόντος στο αρχείο με βάση τα ορίσματα που θα δέχεται.

Να χρησιμοποιηθεί ο παρακάτω κώδικας για το πρόγραμμα:

https://github.com/vasnastos/PROGRAMMING-TO-C-2/blob/master/Course4/%CE%91%CE%A1%CE%A7%CE%95%CE%99%CE%91_EXTRA_%CE%9C%CE%91%CE%98%CE%97%CE%9C%CE%91/%CE%91%CE%A3%CE%9A%CE%97%CE%A3%CE%97_4/cargobase.c