



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Πτυχιακή Εργασία

Τίτλος Πτυχιακής Εργασίας	(Ελληνικά) Οπτικοποίηση Αλγορίθμων για εκπαιδευτικούς σκοπούς (Αγγλικά) Visualization of algorithms for educational purposes
Όνοματεπώνυμο Φοιτητή	ΚΑΛΔΙΡΗ ΑΓΓΕΛΙΚΗ
Πατρώνυμο	ΚΑΛΔΙΡΗΣ ΚΩΝΣΤΑΝΤΙΝΟΣ
Αριθμός Μητρώου	Π20069
Επιβλέπων	ΤΣΑΚΩΝΑΣ ΠΑΝΑΓΙΩΤΗΣ Ε.ΔΙ.Π

Ημερομηνία Παράδοσης

Ιούλιος 2024

Copyright ©

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν αποκλειστικά τον συγγραφέα και δεν αντιπροσωπεύουν τις επίσημες θέσεις του Πανεπιστημίου Πειραιώς.

Ως συγγραφέας της παρούσας εργασίας δηλώνω πως η παρούσα εργασία δεν αποτελεί προϊόν λογοκλοπής και δεν περιέχει υλικό από μη αναφερόμενες πηγές.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου Τσάκωνα Παναγιώτης, μέλος του εργαστηριακού διδακτικού προσωπικού του τμήματος πληροφορικής του πανεπιστημίου Πειραιώς, για την καθοδήγηση που μου προσέφερε και το χρόνο που διέθεσε δίνοντάς μου χρήσιμες συμβουλές και οδηγίες για την ολοκλήρωση της πτυχιακής μου εργασίας.

Περίληψη

Η παρούσα πτυχιακή παροτρύνει το καθένα να κατανοήσει ορισμένες έννοιες των αλγορίθμων μέσα από ασκήσεις οι οποίες δίνονται. Με λίγα λόγια ο αλγόριθμος είναι μία διαδικασία με την οποία λύνεται κάποιο πρόβλημα. Αποτελείται από μία σειρά συγκεκριμένων οδηγιών οι οποίες εκτελούνται βήμα βήμα. Για αυτό το λόγω και στα παραδείγματα τα οποία δίνονται έχουν συγκεκριμένα βήματα τα οποία πρέπει να ακολουθήσει κανείς ώστε να καταφέρει να επιλύσει τα προβλήματα.

Οι αλγόριθμοι οι οποίοι παρουσιάζονται είναι η ταξινόμηση φυσαλίδας (bubblesort), η ταξινόμηση με επιλογή (selection sort) και η γραμμική αναζήτηση (linear search). Αυτοί είναι μερικοί από τους πιο βασικούς αλγόριθμους.

Λέξεις Κλειδιά: αλγόριθμος, πρόβλημα, βήματα, ταξινόμηση φυσαλίδας, ταξινόμηση με επιλογή, γραμμική αναζήτηση

Abstract

This thesis encourages everyone to understand certain concepts of algorithms through exercises which are given. In a nutshell, an algorithm is a procedure by which a problem is solved. It consists of a series of specific instructions which are executed step by step. For this reason the examples which are given have specific steps which one has to follow in order to solve the problems.

The algorithms which are presented are bubblesort, selection sort and linear search. These are some of the most basic algorithms.

Key Words: algorithm, problem, steps, bubblesort, selection sort, linear search

Πίνακας Περιεχομένων

Πτυχιακή Εργασία	i
Copyright ©	i
Ευχαριστίες	ii
Περίληψη	iii
Λέξεις Κλειδιά:.....	iii
Abstract	iii
Key Words:.....	iii
Πίνακας Περιεχομένων	iv
Κατάλογος Εικόνων	v
Εισαγωγή	1
1. Αλγόριθμοι.....	2
1.1 Ταξινόμηση Φυσαλίδας	2
1.2 Ταξινόμηση με επιλογή.....	2
1.3 Γραμμική Αναζήτηση.....	3
2. Ασκήσεις.....	5
Συμπεράσματα	7
Βιβλιογραφία	7
Παράρτημα	7
HTML	7
JavaScript.....	13

Κατάλογος Εικόνων

Ψευδοκώδικας

```
n=arr.length;
for(i=0;i<n-1;i++){
  for(j=0;j<n-i-1;j++){
    if(arr[j]>arr[j+1]){
      temp=arr[j];
      arr[j]=arr[j+1];
      arr[j+1]=temp;
    }
  }
}
```

Το σχήμα 1 παρουσιάζει το ψευδοκώδικα της ταξινόμησης φυσαλίδας.

Ψευδοκώδικας

```
n=arr.length;
for(i=0;i<=n-1;i++){
  sub=i;
  for(j=i+1;j<=n-1;j++){
    if(arr[j]<arr[sub]){
      sub=j;
    }
  }
  temp=arr[i];
  arr[i]=arr[sub];
  arr[sub]=temp;
}
```

Το σχήμα 2 παρουσιάζει το ψευδοκώδικα της ταξινόμησης με επιλογή.

Ψευδοκώδικας

```
key  
n=arr.length;  
for(i=0;i<n;i++){  
    if(arr[i]==key){  
        return i;  
    }  
}
```

Το σχήμα 3 παρουσιάζει το ψευδοκώδικα της γραμμικής αναζήτησης.

Εισαγωγή

Η παρούσα πτυχιακή προβάλλει ορισμένους αλγορίθμους ταξινόμησης και αναζήτησης μέσω ορισμένων ασκήσεων. Ο σκοπός των ασκήσεων είναι να κατανοήσει ο καθένας το τρόπο με τον οποίο μπορούν να επιλυθούν ορισμένα προβλήματα. Οι ασκήσεις πραγματοποιούνται δίνοντας στο πρόγραμμα ορισμένα στοιχεία ώστε να επιλυθούν είτε με πλήρη εκτέλεση είτε με βηματική εκτέλεση.

1. Αλγόριθμοι

1.1 Ταξινόμηση Φυσαλίδας

Με τη ταξινόμηση φυσαλίδας συγκρίνουμε και ανταλλάσσουμε ζεύγη γειτονικών στοιχείων μέχρις ότου να ταξινομηθούν όλα. Με τη μέθοδο αυτή κάθε φορά γίνονται διαδοχικές προσπελάσεις στο πίνακα έτσι ώστε το μικρότερο στοιχείο της ακολουθίας να μετακινηθεί προς το αριστερό άκρο του πίνακα.

Ο ψευδοκώδικας που χρησιμοποιείται είναι ο εξής:

```
n=arr.length;
for(i=0;i<n-1;i++){
    for(j=0;j<n-i-1;j++){
        if(arr[i]>arr[j+1]){
            temp=arr[j];
            arr[j]=arr[j+1];
            arr[j+1]=temp;
        }
    }
}
```

1.2 Ταξινόμηση με επιλογή

Η ταξινόμηση με επιλογή βασίζεται σε τρία βήματα. Το πρώτο βήμα είναι η επιλογή του ελάχιστου στοιχείου. Το δεύτερο βήμα είναι η ανταλλαγή με το i -οστό στοιχείο (i είναι μία μεταβλητή που αυξάνεται κατά ένα). Το τρίτο βήμα είναι η επανάληψη των βημάτων 1 και 2 για τα υπόλοιπα στοιχεία. Ο χρόνος εκτέλεσης της ταξινόμησης με επιλογή είναι

$$\sum_{i=1}^{n-1} i = O(n^2)$$

Ο ψευδοκώδικας που χρησιμοποιείται είναι ο εξής:

```
n=arr.length;
for(i=0;i<=n-1;i++)
    sub=i;
    for(j=i+1;j<=n-1;j++){
        if(arr[j]<arr[sub]){
            sub=j;
        }
    }
temp=arr[i];
arr[i]=arr[sub];
arr[sub]=temp;
```

1.3 Γραμμική Αναζήτηση

Η γραμμική αναζήτηση είναι ένας αλγόριθμος αναζήτησης ενός στοιχείου σε μία δομή δεδομένων. Ο αλγόριθμος χρησιμοποιεί τη μέθοδο της τυφλής αναζήτησης καθώς δεν επιλέγει με κάποια κριτήρια μια υπο-περιοχή αναζήτησης αλλά στηρίζεται στο γεγονός ότι αν ελέγξει όλα τα στοιχεία της δομής δεδομένων ένα προς ένα τότε αν υπάρχει το στοιχείο κλειδί στη δομή θα το βρει. Είναι ο πιο απλός αλγόριθμος αναζήτησης και ο λιγότερος αποδοτικός. Ωστόσο είναι ο απαραίτητος αν η δομή δεν είναι ταξινομημένη.

Η πολυπλοκότητά του είναι $\Theta(n)$.

Ο ψευδοκώδικας που χρησιμοποιείται είναι ο εξής:

```
key;  
n=arr.length;  
for(i=0;i<n;i++){  
    if(arr[i]==key){  
        return i;  
    }  
}
```

2. Ασκήσεις

Ο χρήστης δίνει σαν είσοδο τα στοιχεία της λίστας χωρισμένα με κόμμα και έπειτα του δίνεται η επιλογή να επιλέξει είτε το κουμπί πλήρης εκτέλεση είτε βηματική εκτέλεση.

Με το κουμπί πλήρης εκτέλεση:

Ταξινόμηση Φυσαλίδας (Bubblesort)

Τι είναι η ταξινόμηση φυσαλίδας;
Δώσε σαν είσοδο τα στοιχεία της λίστας χωρισμένα με κόμμα (,) και πάτησε το κουμπί **Πλήρης εκτέλεση** για να δεις όλα τα βήματα ή **Βηματική Εκτέλεση** για να εκτελείς τον αλγόριθμο βήμα βήμα.

Μέγεθος Γραμματοσειράς: Μικρή Μεσαία Μεγάλη

Ψευδοκώδικας

```
n=arr.length;
for(i=0;i<n-1;i++){
  for(j=0;j<n-i-1;j++){
    if(arr[j]>arr[j+1]){
      temp=arr[j];
      arr[j]=arr[j+1];
      arr[j+1]=temp;
    }
  }
}
```

Τιμές της Λίστας

Παρακάτω θα εμφανίζεται η λίστα σε κάθε επανάληψη του αλγορίθμου.
Για κάθε ζεύγος τιμών που αλλάζουν θέση, τα στοιχεία θα χρωματίζονται **κόκκινα**.

Αρχική Λίστα: 5,1,84,31,7,12,3

(i=0,j=0) 1 5 84 31 7 12 3
(i=0,j=1) 1 5 84 31 7 12 3
(i=0,j=2) 1 5 31 84 7 12 3
(i=0,j=3) 1 5 31 7 84 12 3
(i=0,j=4) 1 5 31 7 12 84 3
(i=0,j=5) 1 5 31 7 12 3 84
(i=1,j=0) 1 5 31 7 12 3 84
(i=1,j=1) 1 5 31 7 12 3 84
(i=1,j=2) 1 5 7 31 12 3 84
(i=1,j=3) 1 5 7 12 31 3 84
(i=1,i=4) 1 5 7 12 3 31 84

Με το κουμπί βηματική εκτέλεση:

Γραμμική Αναζήτηση (Linear Search)

Τι είναι η γραμμική αναζήτηση;
Δώσε σαν είσοδο τα στοιχεία της λίστας χωρισμένα με κόμμα (,) και το στοιχείο-κλειδί και πάτησε το κουμπί **Πλήρης εκτέλεση** για να δεις όλα τα βήματα ή **Βηματική Εκτέλεση** για να εκτελείς τον αλγόριθμο βήμα βήμα.

Μέγεθος Γραμματοσειράς: Μικρή Μεσαία Μεγάλη

Ψευδοκώδικας

```
key=12;
n(7)=arr.length;
for(i=0;i<(7)n;i++){
  if(arr[i]==(12)key){
    return i;
  }
}
```

Τρέχουσα Τιμή Αναζήτησης

Σε κάθε επανάληψη θα εμφανίζεται το τρέχον στοιχείο που εξετάζεται με **Κόκκινο**.
Εάν το στοιχείο είναι το στοιχείο-κλειδί θα εμφανιστεί **Πορτοκαλί**.

Αρχική Λίστα: 12, 45, 6, 78, 3, 45, 9
Στοιχείο-Κλειδί: 12

Ψευδοκώδικας

```
key=12;  
n(7)=arr.length;  
for(i=1;i<(7)n;i++){  
    if(arr[i] == 12) return i;  
    return i(12);  
}  
}
```

Τρέχουσα Τιμή Αναζήτησης

Σε κάθε επανάληψη θα εμφανίζεται το τρέχον στοιχείο που εξετάζεται με **Κόκκινο**.

Εάν το στοιχείο είναι το στοιχείο-κλειδί θα εμφανιστεί **Πορτοκαλί**.

Αρχική Λίστα: 12, 45, 6, 78, 3, 45, 9

Στοιχείο-Κλειδί: 12

(i=0) 12, 45, 6, 78, 3, 45, 9

Θέση Στοιχείου: i=0

Συμπεράσματα

Ο χρήστης εφόσον έχει κατανοήσει πλήρως τη θεωρία αυτών των αλγορίθμων μπορεί να πειραματιστεί με διάφορες λίστες με σκοπό να βελτιωθεί αλλά και να μάθει αυτούς τους αλγόριθμους καλύτερα. Ο πειραματισμός με τη βοήθεια των ασκήσεων βοηθά το χρήστη να παρακολουθήσει αναλυτικότερα τα βήματα τα οποία χρειάζονται ώστε να μπορέσει έπειτα να λύνει τα προβλήματα χωρίς καμία βοήθεια.

Στη σημερινή εποχή, οι άνθρωποι πιο δύσκολα θα επιλύσουν προβλήματα στο χαρτί, διότι έχουν συνηθίσει να αναζητούν τις λύσεις από τον υπολογιστή. Επομένως, η ιστοσελίδα βοηθά το χρήστη να σκεφτεί αλλά και να πειραματιστεί.

Βιβλιογραφία

http://repfiles.kallipos.gr/html_books/4410/Ch7.html

<https://aepp.gr/paradigma-taxinomisi-efthias-antallagis/>

Παράρτημα

HTML

- homepage.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Κεντρική Σελίδα</title>
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
    <script src="bubblesort_script.js" type="text/javascript"></script>
  </head>
  <body style="text-align: center;font-size: 22px;background-color: #f7b5c0;
margin-left: 15%;margin-right: 15%;onload="load_page();">
    <div><h1>Προσομοίωση Αλγορίθμων</h1></div><br>
    <div>
      <a href="bubblesort.html">Ταξινόμηση Φυσαλίδας</a><br><br>
      <a href="selection_sort.html">Ταξινόμηση με Επιλογή</a><br><br>
      <a href="linear_search.html">Γραμμική Αναζήτηση</a>
    </div>
  </body>
</html>
```

- bubblesort.html

```
<!DOCTYPE html>
```

```

<html>
  <head>
    <title>Ταξινόμηση Φυσαλίδας</title>
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
    <script src="bubblesort_script.js" type="text/javascript"></script>
  </head>
  <body style="text-align: center;font-size: 20px;background-color: #f7b5c0;
margin-left: 15%;margin-right: 15%;onload="load_page();">
  <div>
    <h1>Ταξινόμηση Φυσαλίδας (Bubblesort)</h1><a id="show_explanation"
onclick="explain();">Τι είναι η ταξινόμηση φυσαλίδας;</a><br>
    <p id="explanation" hidden="true">Η <b>Ταξινόμηση ευθείας ανταλλαγής ή
φυσαλίδας</b> είναι μία μέθοδος ταξινόμησης στην οποία συγκρίνουμε και
ανταλλάσσουμε <b>ζεύγη γειτονικών στοιχείων</b>.<br>
    Με αυτή την μέθοδο κάθε φορά γίνονται διαδοχικές προσπελάσεις στον
πίνακα έτσι ώστε το μικρότερο στοιχείο της ακολουθίας να μετακινηθεί προς το
άκρο του πίνακα που αντιστοιχεί στην μικρότερη τιμή.<br>
    </p>
    <p>
    Δώσε σαν είσοδο τα στοιχεία της λίστας χωρισμένα με κόμμα (,) και
πάτησε το κουμπί <b>Πλήρης εκτέλεση</b> για να δείς όλα τα βήματα ή
<b>Βηματική Εκτέλεση</b> για να εκτελείς τον αλγόριθμο βήμα βήμα.
    </p>
  </div>
  <div>
    <input type="text" size="50" placeholder="Αρχική Λίστα"
id="list_input"><br><br>
    <button onclick="full_execute();">Πλήρης
Εκτέλεση</button>&nbsp;&nbsp;&nbsp;<button onclick="next_step();"
id="step_by_step">Βηματική Εκτέλεση</button>&nbsp;&nbsp;&nbsp;<button
onclick="clear_page();">Καθαρισμός Σελίδας</button>
  </div><br>
  <div>Μέγεθος Γραμματοσειράς:&nbsp;&nbsp;&nbsp;<input type="radio" id="small"
name="font_size" onclick="small();"><label for="small"
>Μικρή</label>&nbsp;&nbsp;&nbsp;<input type="radio" id="medium" name="font_size"
onclick="medium();"><label for="medium">Μεσαία</label>&nbsp;&nbsp;&nbsp;<input
type="radio" id="big" name="font_size" onclick="large();"><label
for="big">Μεγάλη</label></div><br>
  <div>
    <label id="current_list" hidden="true"></label>
    <table id="pseudocode" style="font-size: 25px;background-color:
white;width:35%;text-align: left;float:left;" align="center">
      <tr><td align="center"><b>Ψευδοκώδικας</b></td></tr>
      <tr id="n_set"><td>&nbsp;&nbsp;&nbsp;n<label id="n"></label>=arr.length;</td></tr>
      <tr id="i_plus"><td>&nbsp;&nbsp;&nbsp;for(i=<label id="i"></label>;i<label
id="max_i"></label>;i++){</td></tr>

```



```

        <table style="font-size: 25px;background-color:
white;width:60%;text-align: left;float:right; align="right" hidden="true"
id="values_table">
        </table>
    </div>
</body>
</html>

```

- linear_search.html

```

<!DOCTYPE html>
<html>
    <head>
        <title>Γραμμική Αναζήτηση</title>
        <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
        <script src="linear_search.js" type="text/javascript"></script>
    </head>
    <body style="text-align: center;font-size: 20px;background-color: #f7b5c0;
margin-left: 15%;margin-right: 15%;" onload="load_page();">
        <div><h1>Γραμμική Αναζήτηση (Linear Search)</h1></div>
        <a id="show_explanation" onclick="explain();">Τι είναι η γραμμική
αναζήτηση;</a>
        <p id="explanation" hidden="true">Η γραμμική (linear) ή σειριακή
(sequential) αναζήτηση είναι η πιο απλή μορφή αναζήτησης.<br>
        Ο αλγόριθμος χρησιμοποιεί τη μέθοδο της τυφλής αναζήτησης (brute-
force), καθώς δεν επιλέγει με κάποια κριτήρια κάποια υπο-περιοχή αναζήτησης,
<br>
        αλλά στηρίζεται στο γεγονός ότι εάν ελέγξει όλα τα στοιχεία της
δομής δεδομένων ένα προς ένα, τότε εάν υπάρχει το στοιχείο-κλειδί στην λίστα
θα το βρεί.
        </p>
        <p>
        Δώσε σαν είσοδο τα στοιχεία της λίστας χωρισμένα με κόμμα (,) και
το στοιχείο-κλειδί και πάτησε το κουμπί <b>Πλήρης εκτέλεση</b> για να δείς
όλα τα βήματα ή <b>Βηματική Εκτέλεση</b> για να εκτελείς τον αλγόριθμο βήμα
βήμα.
        </p>
        <div>
            <input type="text" size="30" placeholder="Αρχική Λίστα"
id="list_input">&nbsp;&nbsp;<input type="text" size="15"
placeholder="Στοιχείο-Κλειδί" id="key_element"><br><br>
            <button onclick="full_execute();">Πλήρης
Εκτέλεση</button>&nbsp;&nbsp;<button onclick="next_step();"
id="step_by_step">Βηματική Εκτέλεση</button>&nbsp;&nbsp;<button
onclick="clear_page();">Καθαρισμός Σελίδας</button>
        </div><br>

```


JavaScript

- bubblesort_script.js

```
function load_page() {
    document.getElementById("n").innerHTML = "";
    document.getElementById("i").innerHTML = "0";
    document.getElementById("max_i").innerHTML = "n-1";
    document.getElementById("j").innerHTML = "0";
    document.getElementById("max_j").innerHTML = "n-i-1";
    document.getElementById("arr_j").innerHTML = "";
    document.getElementById("arr_j_1").innerHTML = "";
    document.getElementById("arr_j_temp").innerHTML = "";
    document.getElementById("arr_j_new").innerHTML = "";
    document.getElementById("temp").innerHTML = "";
    document.getElementById("current_list").innerHTML = "-";
    document.getElementById("medium").checked=true;
    medium();
    var table=document.getElementById("values_table");
    table.hidden=true;
    document.getElementById("step_by_step").disabled=false;
    document.getElementById("list_input").disabled=false;
    row=table.insertRow(-1);
    cell=row.insertCell(0);
    cell.innerHTML="<b>Τιμές της Λίστας</b>";
    row=table.insertRow(-1);
    cell=row.insertCell(0);
    cell.innerHTML='Παρακάτω θα εμφανίζεται η λίστα σε κάθε επανάληψη του
αλγορίθμου.<br>Για κάθε ζεύγος τιμών που αλλάζουν θέση, τα στοιχεία θα
χρωματίζονται <b style="color:red;">κόκκινα</b>.';
}

function clear_page(){
    document.getElementById("list_input").value="";
    location.reload();
}

function check_input(){
    var arr_str=document.getElementById("list_input").value;
    var list=arr_str.split(",");
    for(let i=0;i<list.length;i++){
        if(!Number(list[i])){
            return false;
        }
    }
    return true;
}
```

```

function explain(){
    document.getElementById("explanation").hidden=false;
    document.getElementById("show_explanation").hidden=true;
}

function next_step() {
    clear_colors();
    if (check_input()) {
        var table = document.getElementById("values_table");
        var row;
        document.getElementById("list_input").disabled=true;
        if (document.getElementById("current_list").innerHTML == "-") {
            document.getElementById("current_list").innerHTML =
document.getElementById("list_input").value;
            document.getElementById("step_by_step").click();
            document.getElementById("values_table").hidden = false;
            var row = table.insertRow(-1);
            var cell = row.insertCell(0);
            cell.innerHTML = "<br>";
            row = table.insertRow(-1);
            var cell = row.insertCell(0);
            let
str_list=document.getElementById("current_list").innerHTML.split(",");
            list=[];
            for(var i=0;i<str_list.length;i++){
                list.push(Number(str_list[i]));
            }
            cell.innerHTML = "<b>Αρχική Λίστα: " + new_list(list,-2) + "</b>";
        } else {
            var n = document.getElementById("n").innerHTML;
            var array = document.getElementById("current_list").innerHTML;
            array = array.split(",");
            if (n == "") {
                n = array.length;
                document.getElementById("n").innerHTML ="(" + n+")";
                document.getElementById("max_i").innerHTML = n - 1;
                document.getElementById("n_set").style.backgroundColor="yellow
";
            } else {
                i = document.getElementById("max_i").innerHTML;
                if (Number(i)) {
                    document.getElementById("max_j").innerHTML =
Number(document.getElementById("n").innerHTML.replace(",","").replace(")",""))
- Number(document.getElementById("i").innerHTML) - 1;
                    if (Number(document.getElementById("j").innerHTML) >=
Number(document.getElementById("max_j").innerHTML) &&

```

```

Number(document.getElementById("i").innerHTML) <
Number(document.getElementById("max_i").innerHTML)) {
    document.getElementById("j").innerHTML = 0;
    document.getElementById("i").innerHTML =
Number(document.getElementById("i").innerHTML) + 1;
    document.getElementById("i_plus").style.backgroundColo
r="yellow";
    } else if (Number(document.getElementById("i").innerHTML)
< Number(document.getElementById("max_i").innerHTML)) {
    document.getElementById("arr_j").innerHTML =
"+"+array[Number(document.getElementById("j").innerHTML)]+"";
    document.getElementById("arr_j_1").innerHTML =
"+"+array[Number(document.getElementById("j").innerHTML) + 1]+"";
    document.getElementById("j_plus").style.backgroundColo
r="yellow";
    if
(Number(document.getElementById("arr_j").innerHTML.replace("(", "").replace(")", "")) >
Number(document.getElementById("arr_j_1").innerHTML.replace("(", "").replace(")", ""))) {
        document.getElementById("if_condition").style.back
groundColor="green";
        document.getElementById("arr_j_temp").innerHTML =
"+"+array[Number(document.getElementById("j").innerHTML)]+"";
        document.getElementById("arr_j_new").innerHTML =
"+"+array[Number(document.getElementById("j").innerHTML) + 1]+"";
        document.getElementById("temp").innerHTML =
document.getElementById("arr_j_temp").innerHTML;
        array[Number(document.getElementById("j").innerHTM
L)] =
document.getElementById("arr_j_new").innerHTML.replace("(", "").replace(")", "");
        ;
        array[Number(document.getElementById("j").innerHTM
L) + 1] =
document.getElementById("temp").innerHTML.replace("(", "").replace(")", "");
        document.getElementById("tmp_change_1").style.back
groundColor="yellow";
        document.getElementById("tmp_change_2").style.back
groundColor="yellow";
        document.getElementById("tmp_change_3").style.back
groundColor="yellow";
        document.getElementById("current_list").innerHTML
= array;
        var row = table.insertRow(-1);
        var cell = row.insertCell(0);
        let
str_list=document.getElementById("current_list").innerHTML.split(",");
        list=[];

```

```

        for(var i=0;i<str_list.length;i++){
            list.push(Number(str_list[i]));
        }
        cell.innerHTML = '(i=' +
document.getElementById("i").innerHTML + ',j=' +
document.getElementById("j").innerHTML + ') ' +
new_list(list,Number(document.getElementById("j").innerHTML) ,true);
    } else {
        document.getElementById("if_condition").style.back
groundColor="red";

        let
str_list=document.getElementById("current_list").innerHTML.split(",");
        list=[];
        for(var i=0;i<str_list.length;i++){
            list.push(Number(str_list[i]));
        }
        var row = table.insertRow(-1);
        var cell = row.insertCell(0);
        cell.innerHTML = '(i=' +
document.getElementById("i").innerHTML + ',j=' +
document.getElementById("j").innerHTML + ') ' +
new_list(list,Number(document.getElementById("j").innerHTML),false);
    }
        document.getElementById("j").innerHTML =
Number(document.getElementById("j").innerHTML) + 1;
        if (Number(document.getElementById("i").innerHTML) + 1
== Number(document.getElementById("max_i").innerHTML)) {
            var row = table.insertRow(-1);
            var cell = row.insertCell(0);
            let
str_list=document.getElementById("current_list").innerHTML.split(",");
            list=[];
            for(var i=0;i<str_list.length;i++){
                list.push(Number(str_list[i]));
            }
            cell.innerHTML = '<b>Τελική Λίστα: ' +
new_list(list,-2,true) + '</b>';
            document.getElementById("step_by_step").disabled=t
rue;

        }
    }
}
}
}
}
document.getElementById("results_div").scrollTop =
document.getElementById("results_div").scrollHeight;
}else{
    alert("Μη αποδεκτή είσοδος");
}

```

```

    }
}

function full_execute(){
    if(check_input()){
        document.getElementById("values_table").hidden=false;
        clear_colors();
        document.getElementById("if_condition").style.backgroundColor="white";
        document.getElementById("values_table").innerHTML="";
        load_page();
        document.getElementById("step_by_step").disabled=true;
        var array = document.getElementById("list_input").value;
        var table=document.getElementById("values_table");
        table.hidden=false;
        array=array.split(",");
        var numbers=[];

        for(var i=0;i<array.length;i++){
            numbers.push(Number(array[i]));
        }
        var row = table.insertRow(-1);
        var cell = row.insertCell(0);
        cell.innerHTML = "<br>";
        row = table.insertRow(-1);
        var cell = row.insertCell(0);
        cell.innerHTML = "<b>Αρχική Λίστα: " + numbers + "</b>";
        n=numbers.length;
        for(var i=0;i<n-1;i++){
            for(var j=0;j<n-i-1;j++){
                if(numbers[j]>numbers[j+1]){
                    temp=numbers[j];
                    numbers[j]=numbers[j+1];
                    numbers[j+1]=temp;
                    row = table.insertRow(-1);
                    var cell = row.insertCell(0);
                    cell.innerHTML = '(i='+i+',j='+j+') '
+new_list(numbers,j,true) ;
                }else{
                    row = table.insertRow(-1);
                    var cell = row.insertCell(0);
                    cell.innerHTML = '(i='+i+',j='+j+') ' +
new_list(numbers,j,false) + '</b>';
                }
            }
        }
        row = table.insertRow(-1);
        var cell = row.insertCell(0);
        cell.innerHTML = '<b>Τελική Λίστα: ' + new_list(numbers,-2) + '</b>';
    }
}

```



```

        document.getElementById("results_div").scrollTop =
document.getElementById("results_div").scrollHeight;
    }else{
        alert("Μη αποδεκτή είσοδος");
    }
}

function new_list(list,j,switched){
    var new_str="";
    if(switched){
        for(var i=0;i<list.length;i++){
            if(i==j || i==j+1){
                new_str=new_str+' <b style="color:red">'+list[i]+'</b>';
            }else{
                new_str=new_str+" "+list[i];
            }
        }
    }else{
        for(var i=0;i<list.length;i++){
            if(i==j){
                new_str=new_str+' <b style="color:red">'+list[i]+'</b>';
            }else{
                new_str=new_str+" "+list[i];
            }
        }
    }
    return new_str;
}

function clear_colors(){
    document.getElementById("n_set").style.backgroundColor="white";
    document.getElementById("i_plus").style.backgroundColor="white";
    document.getElementById("j_plus").style.backgroundColor="white";
    document.getElementById("tmp_change_1").style.backgroundColor="white";
    document.getElementById("tmp_change_2").style.backgroundColor="white";
    document.getElementById("tmp_change_3").style.backgroundColor="white";
}

function small(){
    document.getElementById("values_table").style.fontSize="18px";
    document.getElementById("pseudocode").style.fontSize="18px";
}

function medium(){
    document.getElementById("values_table").style.fontSize="22px";
    document.getElementById("pseudocode").style.fontSize="22px";
}

```

```

}

function large(){
    document.getElementById("values_table").style.fontSize="30px";
    document.getElementById("pseudocode").style.fontSize="30px";
}

```

- selection_sort_script.js

```

function load_page() {
    document.getElementById("n").innerHTML = "";
    document.getElementById("i").innerHTML = "0";
    document.getElementById("max_i").innerHTML = "n-1";
    document.getElementById("j").innerHTML = "0";
    document.getElementById("max_j").innerHTML = "n-i-1";
    document.getElementById("arr_j").innerHTML = "";
    document.getElementById("arr_j_1").innerHTML = "";
    document.getElementById("arr_j_temp").innerHTML = "";
    document.getElementById("arr_j_new").innerHTML = "";
    document.getElementById("temp").innerHTML = "";
    document.getElementById("current_list").innerHTML = "-";
    document.getElementById("medium").checked=true;
    medium();
    var table=document.getElementById("values_table");
    table.hidden=true;
    document.getElementById("step_by_step").disabled=false;
    document.getElementById("list_input").disabled=false;
    row=table.insertRow(-1);
    cell=row.insertCell(0);
    cell.innerHTML="<b>Τιμές της Λίστας</b>";
    row=table.insertRow(-1);
    cell=row.insertCell(0);
    cell.innerHTML='Παρακάτω θα εμφανίζεται η λίστα σε κάθε επανάληψη του
αλγορίθμου.<br>Για κάθε ζεύγος τιμών που αλλάζουν θέση, τα στοιχεία θα
χρωματίζονται <b style="color:red;">κόκκινα</b>.';
}

```

```

function clear_page(){
    document.getElementById("list_input").value="";
    location.reload();
}

```

```

function check_input(){
    var arr_str=document.getElementById("list_input").value;
    var list=arr_str.split(",");
    for(let i=0;i<list.length;i++){
        if(!Number(list[i])){

```

```

        return false;
    }
}
return true;
}

function explain(){
    document.getElementById("explanation").hidden=false;
    document.getElementById("show_explanation").hidden=true;
}

function next_step() {
    clear_colors();
    if (check_input()) {
        var table = document.getElementById("values_table");
        var row;
        document.getElementById("list_input").disabled=true;
        if (document.getElementById("current_list").innerHTML == "-") {
            document.getElementById("current_list").innerHTML =
document.getElementById("list_input").value;
            document.getElementById("step_by_step").click();
            document.getElementById("values_table").hidden = false;
            var row = table.insertRow(-1);
            var cell = row.insertCell(0);
            cell.innerHTML = "<br>";
            row = table.insertRow(-1);
            var cell = row.insertCell(0);
            let
str_list=document.getElementById("current_list").innerHTML.split(",");
            list=[];
            for(var i=0;i<str_list.length;i++){
                list.push(Number(str_list[i]));
            }
            cell.innerHTML = "<b>Αρχική Λίστα: " + new_list(list,-2) + "</b>";
        } else {
            var n = document.getElementById("n").innerHTML;
            var array = document.getElementById("current_list").innerHTML;
            array = array.split(",");
            if (n == "") {
                n = array.length;
                document.getElementById("n").innerHTML = "(" + n + ")";
                document.getElementById("max_i").innerHTML = n - 1;
                document.getElementById("n_set").style.backgroundColor="yellow
";
            } else {
                i = document.getElementById("max_i").innerHTML;
                if (Number(i)) {

```

```

        document.getElementById("max_j").innerHTML =
Number(document.getElementById("n").innerHTML.replace(", "").replace(".", ""))
- Number(document.getElementById("i").innerHTML) - 1;
        if (Number(document.getElementById("j").innerHTML) >=
Number(document.getElementById("max_j").innerHTML) &&
Number(document.getElementById("i").innerHTML) <
Number(document.getElementById("max_i").innerHTML)) {
            document.getElementById("j").innerHTML = 0;
            document.getElementById("i").innerHTML =
Number(document.getElementById("i").innerHTML) + 1;
            document.getElementById("i_plus").style.backgroundColo
r="yellow";
        } else if (Number(document.getElementById("i").innerHTML)
< Number(document.getElementById("max_i").innerHTML)) {
            document.getElementById("arr_j").innerHTML =
"+"+array[Number(document.getElementById("j").innerHTML)]+"";
            document.getElementById("arr_j_1").innerHTML =
"+"+array[Number(document.getElementById("j").innerHTML) + 1]+"";
            document.getElementById("j_plus").style.backgroundColo
r="yellow";
            if
(Number(document.getElementById("arr_j").innerHTML.replace(", ").replace(")",
"")) >
Number(document.getElementById("arr_j_1").innerHTML.replace(", ").replace(")",
""))) {
                document.getElementById("if_condition").style.back
groundColor="green";
                document.getElementById("arr_j_temp").innerHTML =
"+"+array[Number(document.getElementById("j").innerHTML)]+"";
                document.getElementById("arr_j_new").innerHTML =
"+"+array[Number(document.getElementById("j").innerHTML) + 1]+"";
                document.getElementById("temp").innerHTML =
document.getElementById("arr_j_temp").innerHTML;
                array[Number(document.getElementById("j").innerHTM
L)] =
document.getElementById("arr_j_new").innerHTML.replace(", ").replace(")", "");
                ;
                array[Number(document.getElementById("j").innerHTM
L) + 1] =
document.getElementById("temp").innerHTML.replace(", ").replace(")", "");
                document.getElementById("tmp_change_1").style.back
groundColor="yellow";
                document.getElementById("tmp_change_2").style.back
groundColor="yellow";
                document.getElementById("tmp_change_3").style.back
groundColor="yellow";
                document.getElementById("current_list").innerHTML
= array;

```

```

        var row = table.insertRow(-1);
        var cell = row.insertCell(0);
        let
str_list=document.getElementById("current_list").innerHTML.split(",");
        list=[];
        for(var i=0;i<str_list.length;i++){
            list.push(Number(str_list[i]));
        }
        cell.innerHTML = '(i=' +
document.getElementById("i").innerHTML + ',j=' +
document.getElementById("j").innerHTML + ') ' +
new_list(list,Number(document.getElementById("j").innerHTML) ,true);
    } else {
        document.getElementById("if_condition").style.back
groundColor="red";

        let
str_list=document.getElementById("current_list").innerHTML.split(",");
        list=[];
        for(var i=0;i<str_list.length;i++){
            list.push(Number(str_list[i]));
        }
        var row = table.insertRow(-1);
        var cell = row.insertCell(0);
        cell.innerHTML = '(i=' +
document.getElementById("i").innerHTML + ',j=' +
document.getElementById("j").innerHTML + ') ' +
new_list(list,Number(document.getElementById("j").innerHTML),false);
    }
    document.getElementById("j").innerHTML =
Number(document.getElementById("j").innerHTML) + 1;
    if (Number(document.getElementById("i").innerHTML) + 1
== Number(document.getElementById("max_i").innerHTML)) {
        var row = table.insertRow(-1);
        var cell = row.insertCell(0);
        let
str_list=document.getElementById("current_list").innerHTML.split(",");
        list=[];
        for(var i=0;i<str_list.length;i++){
            list.push(Number(str_list[i]));
        }
        cell.innerHTML = '<b>Τελική Λίστα: ' +
new_list(list,-2,true) + '</b>';
        document.getElementById("step_by_step").disabled=t
rue;
    }
}
}
}
}

```

```

    }
    document.getElementById("results_div").scrollTop =
document.getElementById("results_div").scrollHeight;
    }else{
        alert("Μη αποδεκτή είσοδος");
    }
}

function full_execute(){
    if(check_input()){
        document.getElementById("values_table").hidden=false;
        clear_colors();
        document.getElementById("if_condition").style.backgroundColor="white";
        document.getElementById("values_table").innerHTML="";
        load_page();
        document.getElementById("step_by_step").disabled=true;
        var array = document.getElementById("list_input").value;
        var table=document.getElementById("values_table");
        table.hidden=false;
        array=array.split(",");
        var numbers=[];

        for(var i=0;i<array.length;i++){
            numbers.push(Number(array[i]));
        }
        var row = table.insertRow(-1);
        var cell = row.insertCell(0);
        cell.innerHTML = "<br>";
        row = table.insertRow(-1);
        var cell = row.insertCell(0);
        cell.innerHTML = "<b>Αρχική Λίστα: " + numbers + "</b>";
        n=numbers.length;
        for(var i=0;i<n-1;i++){
            for(var j=0;j<n-i-1;j++){
                if(numbers[j]>numbers[j+1]){
                    temp=numbers[j];
                    numbers[j]=numbers[j+1];
                    numbers[j+1]=temp;
                    row = table.insertRow(-1);
                    var cell = row.insertCell(0);
                    cell.innerHTML = '(i='+i+',j='+j+' ) '
+new_list(numbers,j,true) ;
                }else{
                    row = table.insertRow(-1);
                    var cell = row.insertCell(0);
                    cell.innerHTML = '(i='+i+',j='+j+' ) ' +
new_list(numbers,j,false) + '</b>';
                }
            }
        }
    }
}

```

```

    }
  }
  row = table.insertRow(-1);
  var cell = row.insertCell(0);
  cell.innerHTML = '<b>Τελική Λίστα: ' + new_list(numbers,-2) + '</b>';
  document.getElementById("results_div").scrollTop =
document.getElementById("results_div").scrollHeight;
  }else{
    alert("Μη αποδεκτή είσοδος");
  }
}

```

```

function new_list(list,j,switched){
  var new_str="";
  if(switched){
    for(var i=0;i<list.length;i++){
      if(i==j || i==j+1){
        new_str=new_str+' <b style="color:red">'+list[i]+'</b>';
      }else{
        new_str=new_str+" "+list[i];
      }
    }
  }else{
    for(var i=0;i<list.length;i++){
      if(i==j){
        new_str=new_str+' <b style="color:red">'+list[i]+'</b>';
      }else{
        new_str=new_str+" "+list[i];
      }
    }
  }
  return new_str;
}

```

```

function clear_colors(){
  document.getElementById("n_set").style.backgroundColor="white";
  document.getElementById("i_plus").style.backgroundColor="white";
  document.getElementById("j_plus").style.backgroundColor="white";
  document.getElementById("tmp_change_1").style.backgroundColor="white";
  document.getElementById("tmp_change_2").style.backgroundColor="white";
  document.getElementById("tmp_change_3").style.backgroundColor="white";
}

```

```

function small(){
  document.getElementById("values_table").style.fontSize="18px";
  document.getElementById("pseudocode").style.fontSize="18px";
}

```

```

}

function medium(){
    document.getElementById("values_table").style.fontSize="22px";
    document.getElementById("pseudocode").style.fontSize="22px";
}

function large(){
    document.getElementById("values_table").style.fontSize="30px";
    document.getElementById("pseudocode").style.fontSize="30px";
}

```

- linear_search.js

```

function load_page(){
    clear_colors();
    document.getElementById("key_init").innerHTML="";
    document.getElementById("n_max").innerHTML="";
    document.getElementById("n").innerHTML="";
    document.getElementById("key").innerHTML="";
    document.getElementById("i").innerHTML="0";
    document.getElementById("arr_i").innerHTML="";
    document.getElementById("key_if").innerHTML="";
    document.getElementById("arr_i_return").innerHTML="";
    document.getElementById("current_list").innerHTML="-";
    document.getElementById("key").innerHTML="-";
    document.getElementById("list_input").disabled=false;
    document.getElementById("key_element").disabled=false;
    document.getElementById("step_by_step").disabled=false;
    document.getElementById("medium").checked=true;
    medium();
    var table=document.getElementById("values_table");
    table.hidden=true;
    row=table.insertRow(-1);
    cell=row.insertCell(0);
    cell.innerHTML="<b>Τρέχουσα Τιμή Αναζήτησης</b>";
    row=table.insertRow(-1);
    cell=row.insertCell(0);
    cell.innerHTML='Σε κάθε επανάληψη θα εμφανίζεται το τρέχον στοιχείο που
εξετάζεται με <b style="color:red">Κόκκινο</b>.<br>Εάν το στοιχείο είναι το
στοιχείο-κλειδί θα εμφανιστεί <b
style="color:orange">Πορτοκαλί</b>.<br><br>';}

function next_step(){
    clear_colors();
    if(check_input()==0){
        var table = document.getElementById("values_table");
        var row;
        document.getElementById("list_input").disabled=true;

```



```

document.getElementById("key_element").disabled=true;
table.hidden=false;
if (document.getElementById("current_list").innerHTML == "-") {
    var list = document.getElementById("list_input").value;
    var key = document.getElementById("key_element").value;
    row = table.insertRow(-1);
    cell = row.insertCell(0);
    cell.innerHTML = "<b>Αρχική Λίστα: " + list + "</b>";
    row = table.insertRow(-1);
    cell = row.insertCell(0);
    cell.innerHTML = "<b>Στοιχείο-Κλειδί: " + key + "</b><br><br>";
    document.getElementById("current_list").innerHTML=list;
    document.getElementById("key_element").innerHTML=key;
    document.getElementById("step_by_step").click();
}else{
    var n = document.getElementById("n").innerHTML;
    var array = document.getElementById("current_list").innerHTML;
    array = array.split(",");
    if (n == "") {
        n = array.length;
        document.getElementById("n").innerHTML = "(" + n + ")";
        document.getElementById("n_max").innerHTML = "(" + n + ")";
        document.getElementById("key_init").innerHTML="="+document.get
ElementById("key_element").value+";";
        document.getElementById("key_if").innerHTML="("+document.getEl
ementById("key_element").value+");";
        document.getElementById("set_key").style.backgroundColor="yell
ow";
        document.getElementById("set_n").style.backgroundColor="yellow
";
    }else{
        var array = document.getElementById("current_list").innerHTML;
        array = array.split(",");
        i=Number(document.getElementById("i").innerHTML);
        document.getElementById("arr_i").innerHTML="("+array[i]+)";
        if(array[i]==Number(document.getElementById("key_element").inn
erHTML)){
            document.getElementById("if_condition").style.backgroundCo
lor="green";
            document.getElementById("arr_i_return").innerHTML="("+arra
y[i]+)";
            document.getElementById("step_by_step").disabled=true;
            row = table.insertRow(-1);
            cell = row.insertCell(0);
            cell.innerHTML = "(i="+i+"
"+new_list(document.getElementById("current_list").innerHTML,i,true);
            document.getElementById("set_return_i").style.backgroundCo
lor="orange";

```

```

        row = table.insertRow(-1);
        cell = row.insertCell(0);
        cell.innerHTML = "<br><b> Θέση Στοιχείου: i="+i+"</b>"
    }else{
        document.getElementById("if_condition").style.backgroundCo
lor="red";

        row = table.insertRow(-1);
        cell = row.insertCell(0);
        cell.innerHTML = "(i="+i+"
"+new_list(document.getElementById("current_list").innerHTML,i,false);
    }
    if(i+1<array.length){
        document.getElementById("i").innerHTML=i+1;
        document.getElementById("set_for_i").style.backgroundColor
="yellow";
    }else{
        document.getElementById("step_by_step").disabled=true;
        row = table.insertRow(-1);
        cell = row.insertCell(0);
        cell.innerHTML = "<br><b>Δεν βρέθηκε το στοιχείο-
κλειδί</b>"
    }
    }
}
}
}else if(check_input()==-1){
    alert("Μη αποδεκτή λίστα");
}else{
    alert("Δεν δόθηκε στοιχείο κλειδί");
}
document.getElementById("results_div").scrollTop =
document.getElementById("results_div").scrollHeight;
}

function check_input(){
    var list=document.getElementById("list_input").value.split(",");
    for(var i=0;i<list.length;i++){
        if(!Number(list[i])){
            return -1;
        }
    }
    if(document.getElementById("key_element").value.trim().length===0){
        return -2;
    }
    return 0;
}

function clear_page(){
    document.getElementById("list_input").value="";

```

```

document.getElementById("key_element").value="";
document.getElementById("if_condition").style.backgroundColor="white";
clear_colors();
load_page();
}

function explain(){
document.getElementById("explanation").hidden=false;
document.getElementById("show_explanation").hidden=true;
}

function small(){
document.getElementById("values_table").style.fontSize="18px";
document.getElementById("pseudocode").style.fontSize="18px";
}

function medium(){
document.getElementById("values_table").style.fontSize="22px";
document.getElementById("pseudocode").style.fontSize="22px";
}

function large(){
document.getElementById("values_table").style.fontSize="30px";
document.getElementById("pseudocode").style.fontSize="30px";
}

function new_list(current_list,pos,is_found){
var array=current_list.split(",");
if(pos==0){
if(is_found){
array_string='<b style="color:orange">'+array[0]+'</b>';
}else{
array_string='<b style="color:red">'+array[0]+'</b>';
}
}else{
array_string='<b>'+array[0]+'</b>';
}
for(var i=1;i<array.length;i++){
if(i==pos){
if(is_found){
array_string=array_string+',<b
style="color:orange">'+array[i]+'</b>';
}else{
array_string=array_string+',<b
style="color:red">'+array[i]+'</b>';
}
}else{
array_string=array_string+',<b>'+array[i]+'</b>';
}
}
}

```

```

    }
  }
  return array_string;
}

function full_execute() {
  if (check_input() == 0) {
    document.getElementById("results_div").scrollTop =
document.getElementById("results_div").scrollHeight;
    var table = document.getElementById("values_table");
    table.innerHTML = "";
    document.getElementById("if_condition").style.backgroundColor =
"white";
    load_page();
    table.hidden = false;
    var list=document.getElementById("list_input").value.split(",");
    var key=document.getElementById("key_element").value;
    document.getElementById("step_by_step").disabled = true;
    for(var i=0;i<list.length;i++){
      if(Number(key)==Number(list[i])){
        row = table.insertRow(-1);
        cell = row.insertCell(0);
        cell.innerHTML = "(i="+i+"
"+new_list(list.toString(),i,true);
        row = table.insertRow(-1);
        cell = row.insertCell(0);
        cell.innerHTML = "<br><b> Θέση Στοιχείου: i="+i+"</b>"
        return;
      }else{
        row = table.insertRow(-1);
        cell = row.insertCell(0);
        cell.innerHTML = "(i="+i+"
"+new_list(list.toString(),i,false);
      }
    }
    row = table.insertRow(-1);
    cell = row.insertCell(0);
    cell.innerHTML = "<br><b>Δεν βρέθηκε το στοιχείο-κλειδί</b>";
  } else if (check_input() == -1) {
    alert("Μη αποδεκτή λίστα");
  } else {
    alert("Δεν δόθηκε στοιχείο κλειδί");
  }
}

function clear_colors(){
  document.getElementById("set_key").style.backgroundColor="white";
  document.getElementById("set_n").style.backgroundColor="white";
}

```

```
document.getElementById("set_for_i").style.backgroundColor="white";  
document.getElementById("set_return_i").style.backgroundColor="white";  
}
```