

ΑΕΜ	ΒΑΘΜΟΣ	ΠΑΡΑΤΗΡΗΣΕΙΣ
1138	PASS	- Απαράδεκτα ονόματα μεταβλητών. Μη χρησιμοποιείς ονόματα του ενός γράμματος γιατί κάνουν ακαταλαβίστικο τον κώδικα. - Αν μια μεταβλητή είναι δείκτης, να τη συγκρίνεις με NULL και όχι με 0
1313	PASS	- Στη γραμμή 264 ελέγχεις την τιμή της line, η οποία όμως δεν έχει αρχικοποιηθεί.
1399	PASS	
1462	PASS	
1473	PASS	
1477	PASS	
1556	PASS	
1636	PASS Προσέξτε τις παρατηρήσεις	- Αν η if της γραμμής 220 είναι ψευδής, τότε η συνάρτηση θα επιστρέψει curf το οποίο δε δείχνει πουθενά (έχει σκουπίδια) γιατί δεν έχει αρχικοποιηθεί. - Η strcpy δεν πρέπει να χρησιμοποιείται όταν αντιγράφει ένα string πάνω στον εαυτό του, γιατί τότε έχει απροσδιόριστη συμπεριφορά. Αλλά ο λόγος που σκάει το πρόγραμμά σας στη γραμμή 192 είναι γιατί βγαίνει εκτός ορίων της μνήμης που έχει δεσμευθεί για τη line. Αρκούσε να ξεκινήσετε από το &line[2] την αναζήτηση. - Ελλιπής αναζήτηση λέξης με καλύτερο σκορ. Κοιπάτε μόνο την πρώτη λέξη κάθε λίστας. Χρειαζόσασταν και μια δεύτερη αναδρομική συνάρτηση που θα έβρισκε την καλύτερη λέξη σε κάθε λίστα. Μετά, ψάχνετε ανάμεσα σε αυτές για να βρείτε τη συνολικά καλύτερη λέξη. - Πριν προσπελάσετε οτιδήποτε μεγαλύτερο του argv[0] πρέπει να ελέγχετε το argc (κάνετε προσπέλαση του argv[2] ακόμη κι όταν το argc είναι 2)
1637	PASS	
1734	FAIL	Το πρόγραμμα δε μεταγλωττίζεται και τα λάθη που έχει είναι σημαντικά.
1759	οριακό PASS. Πρόσεξε τις παρατηρήσεις	- Να δίνεις περιγραφικά ονόματα στις μεταβλητές για να είναι κατανοητός ο κώδικάς σου. Παραδείγματα κακών ονομάτων: gnw, c, - Το j στις γραμμές 287, 333 δεν έχει κάποια τιμή. - Πριν προσπελάσεις οτιδήποτε μεγαλύτερο του argv[0] πρέπει να ελέγχεις το argc (κάνεις προσπέλαση του argv[2] ακόμη κι όταν το argc είναι 2) - Το πρόγραμμα σκάει στη γραμμή 199 (το hash_pos έχει αρνητική τιμή γιατί δεν δηλώθηκε ως unsigned και αντί για position που έχει προσαρμοστεί στο μέγεθος του πίνακα χρησιμοποιείς hash_pos) - Το πρόγραμμα σκάει στη γραμμή 333 γιατί γίνεται προσπέλαση του word_b ακόμη κι όταν είναι NULL. - Το πρόγραμμα σκάει στη γραμμή 355 γιατί η μεταβλητή user έχει τιμή NULL. - Δεν αποδεσμεύεται η δυναμικά δεσμευμένη μνήμη του προγράμματος - Η βασική περίπτωση στην αναδρομή είναι για head == NULL. Αν ο πρώτος έλεγχος είναι για το head->nxt, τότε αν η λίστα είναι αρχικά άδεια, θα πετάξει segfault. - Το αναδρομικό βήμα στη find_best_score είναι λάθος. Ουσιαστικά προσπαθείς να βρες την καλύτερη λέξη επαναληπτικά (αλλά ακόμη και ο επαναληπτικός αλγόριθμός δεν είναι σωστός).

1864	PASS	- Το max.num_appearances = 10000000000; δεν είναι σωστό γιατί το 10000000000 είναι μεγαλύτερο από τον μέγιστο δυνατό ακέραιο. Επρεπε να είχες χρησιμοποιήσει το INT_MAX το οποίο ορίζεται στο limits.h - Σωστά λειτουργεί η αναδρομή, αλλά γιατί έχεις δύο κόμβους ως παραμέτρους? Αρκούσε ένας (σκέψου το!)
1868	PASS	Αυτό που κάνετε στη γραμμή 352 λειτουργεί μόνο αν ο word περιέχει ένα ψηφίο. Προφανώς στη συγκεκριμένη άσκηση ισχύει αυτό, αλλά έχετε το υπόψη, κι αν δεν την ξέρετε ήδη δείτε τη συνάρτηση atoi η οποία ενδείκνυται για τέτοιες μετατροπές.
1875	PASS	Αυτό που κάνετε στη γραμμή 352 λειτουργεί μόνο αν ο word περιέχει ένα ψηφίο. Προφανώς στη συγκεκριμένη άσκηση ισχύει αυτό, αλλά έχετε το υπόψη, κι αν δεν την ξέρετε ήδη δείτε τη συνάρτηση atoi η οποία ενδείκνυται για τέτοιες μετατροπές.
2059	PASS	Δεν αποδεσμεύετε τη μνήμη για κάθε μία λέξη που βρίσκεται σε κόμβο λίστας.
2156	PASS Προσέξτε τις παρατηρήσεις	- Το πρόγραμμα σκάει γιατί στις γραμμές 235, 240 γίνεται διαίρεση με το μηδέν. - Πριν προσπελάσετε οτιδήποτε μεγαλύτερο του argv[0] πρέπει να ελέγχετε το argc (κάνετε προσπέλαση του argv[2] ακόμη κι όταν το argc είναι 2) - Στην αναδρομή, σε περίπτωση ισοτιμίας επιστρέφετε την τελευταία λέξη από όσες έχουν μέγιστο σκορ. Σκεφτείτε τι αλλαγή θα έπρεπε να γίνει στον κώδικα ώστε να επιστρέφεται η πρώτη. - Δεν αποδεσμεύετε τη μνήμη για κάθε μία λέξη που βρίσκεται σε κόμβο λίστας. - Στην absolute_word ο πρώτος έλεγχος που έπρεπε να κάνετε είναι για το αν το pos είναι εντός ορίων πίνακα. - Μετά το rehash έπρεπε να αποδεσμεύετε την αρχική μνήμη του πίνακα (το temp δηλαδή).
2157	πολύ οριακό PASS Προσέξτε τις παρατηρήσεις	- Χρειάζεστε αρκετή εξάσκηση ακόμη σε ζητήματα μνήμης και αναδρομής - Όταν σας δίνονται προδιαγραφές για την είσοδο ενός προγράμματος να τις ακολουθείτε. Το πρόγραμμα σκάει όταν το τρέχουμε με τα αρχεία ελέγχου γιατί ζητάτε να διαβαστεί το hashtable.size αντί να του δίνετε εσείς την τιμή που πρέπει. - Δεν επιτρέπεται η χρήση στατικών πινάκων με μεταβλητό μέγεθος. - Στη malloc για τη λέξη δε δεσμεύετε αρκετή μνήμη ώστε να χωρέσει και το '\0' (γρ. 43) - Άσχημα σχεδιασμένη αναδρομική συνάρτηση. Δε γίνεται χρήση του αποτελέσματος που επιστρέφει η αναδρομική κλήση. - Δε γίνεται κλήση της συνάρτησης delete για την αφαίρεση κόμβου. Επίσης, αν το head είναι NULL, θα πετάξει segfault. Η προσπέλαση runner->next->next έχει επίσης πιθανότητα να πετάξει segfault. - Δεν υλοποιήθηκε η rehash.
2165	PASS	Στην αναδρομή, σε περίπτωση ισοτιμίας επιστρέφετε την τελευταία λέξη από όσες έχουν μέγιστο σκορ. Σκεφτείτε τι αλλαγή θα έπρεπε να γίνει στον κώδικα ώστε να επιστρέφεται η πρώτη.
2186	PASS	Στην αναδρομή, σε περίπτωση ισοτιμίας επιστρέφετε την τελευταία λέξη από όσες έχουν μέγιστο σκορ. Σκεφτείτε τι αλλαγή θα έπρεπε να γίνει στον κώδικα ώστε να επιστρέφεται η πρώτη.
2222	FAIL	

2253	PASS Προσέξτε τις παρατηρήσεις	<ul style="list-style-type: none"> <li>- Αν η if της γραμμής 220 είναι ψευδής, τότε η συνάρτηση θα επιστρέψει curr το οποίο δε δείχνει πουθενά (έχει σκουπίδια) γιατί δεν έχει αρχικοποιηθεί.</li> <li>- Η strcpy δεν πρέπει να χρησιμοποιείται όταν αντιγράφει ένα string πάνω στον εαυτό του, γιατί τότε έχει απροσδιόριστη συμπεριφορά. Αλλά ο λόγος που σκάει το πρόγραμμά σας στη γραμμή 192 είναι γιατί βγαίνει εκτός ορίων της μνήμης που έχει δεσμευθεί για τη line. Αρκούσε να ξεκινήσετε από το &amp;line[2] την αναζήτηση.</li> <li>- Ελλιπής αναζήτηση λέξης με καλύτερο σκορ. Κοιτάτε μόνο την πρώτη λέξη κάθε λίστας. Χρειαζόσασταν και μια δεύτερη αναδρομική συνάρτηση που θα έβρισκε την καλύτερη λέξη σε κάθε λίστα. Μετά, ψάχνετε ανάμεσα σε αυτές για να βρείτε τη συνολικά καλύτερη λέξη.</li> <li>- Πριν προσπελάσετε οτιδήποτε μεγαλύτερο του argv[0] πρέπει να ελέγχετε το argc (κάνετε προσπέλαση του argv[2] ακόμη κι όταν το argc είναι 2)</li> </ul>
2259	PASS	<ul style="list-style-type: none"> <li>- Όλος ο κώδικας είναι ένα tab πιο δεξιά από όσο έπρεπε.</li> <li>- Αν η if της γραμμής 256 είναι ψευδής, τότε η γραμμή 290 θα βγάλει λάθος αποτέλεσμα γιατί το i δε θα έχει πάρει τιμή. Έπρεπε να το είχα αρχικοποιήσει καταλλήλως.</li> <li>-Κάπως επικίνδυνες οι συνεχείς προσθαφαιρέσεις στο variable.table Γιατί δε χρησιμοποιείς variable.table[i]?</li> <li>- Σωστό αποτέλεσμα βγάζει η αναδρομή, αλλά μεταβάλλει τα περιεχόμενα της κάθε λίστας. Σκέψου τι αλλαγή θα έπρεπε να κάνεις ώστε να επιστρέφεται ο σωστός κόμβος χωρίς να έχει αλλάξει κάτι στη λίστα.</li> <li>- ο όρος είναι rehash, όχι reharh :)</li> </ul>
2272	PASS	
2275	πολύ οριακό PASS Προσέξτε τις παρατηρήσεις	<ul style="list-style-type: none"> <li>-Χρειάζεστε αρκετή εξάσκηση ακόμη σε ζητήματα μνήμης και αναδρομής</li> <li>- Όταν σας δίνονται προδιαγραφές για την είσοδο ενός προγράμματος να τις ακολουθείτε. Το πρόγραμμα σκάει όταν το τρέχουμε με τα αρχεία ελέγχου γιατί ζητάτε να διαβαστεί το hashtable.size αντί να του δίνετε εσείς την τιμή που πρέπει.</li> <li>- Δεν επιτρέπεται η χρήση στατικών πινάκων με μεταβλητό μέγεθος.</li> <li>- Στη malloc για τη λέξη δε δεσμεύετε αρκετή μνήμη ώστε να χωρέσει και το '\0' (γρ. 43)</li> <li>- Άσχημα σχεδιασμένη αναδρομική συνάρτηση. Δε γίνεται χρήση του αποτελέσματος που επιστρέφει η αναδρομική κλήση.</li> <li>- Δε γίνεται κλήση της συνάρτησης delete για την αφαίρεση κόμβου. Επίσης, αν το head είναι NULL, θα πετάξει segfault. Η προσπέλαση runner-&gt;next-&gt;next έχει επίσης πιθανότητα να πετάξει segfault.</li> <li>- Δεν υλοποιήθηκε η rehash.</li> </ul>
2280	PASS	
2290	PASS	<ul style="list-style-type: none"> <li>- Το πρόγραμμα σκάει γιατί στις γραμμές 235, 240 γίνεται διαίρεση με το μηδέν.</li> <li>- Πριν προσπελάσετε οτιδήποτε μεγαλύτερο του argv[0] πρέπει να ελέγχετε το argc (κάνετε προσπέλαση του argv[2] ακόμη κι όταν το argc είναι 2)</li> <li>- Στην αναδρομή, σε περίπτωση ισοτιμίας επιστρέφετε την τελευταία λέξη από όσες έχουν μέγιστο σκορ. Σκεφτείτε τι αλλαγή θα έπρεπε να γίνει στον κώδικα ώστε να επιστρέφεται η πρώτη.</li> <li>- Δεν αποδεσμεύετε τη μνήμη για κάθε μία λέξη που βρίσκεται σε κόμβο λίστας.</li> <li>- Στην absolute_word ο πρώτος έλεγχος που έπρεπε να κάνετε είναι για το αν το pos είναι εντός ορίων πίνακα.</li> <li>- Μετά το rehash έπρεπε να αποδεσμεύετε την αρχική μνήμη του πίνακα (το temp δηλαδή).</li> </ul>

2300	PASS	- Δεν αποδεσμεύετε τη μνήμη για κάθε μία λέξη που βρίσκεται σε κόμβο λίστας. - Στη max_score, αν η συνθήκη της γραμμής 82 είναι αληθής, τότε επιστρέφετε τη μεταβλητή max η οποία όμως δεν έχει αρχικοποιηθεί (και περιέχει σκουπίδια).
2306	PASS	Η δεύτερη if στην anadromiko_max_list είναι περιττή. Αν το next είναι NULL, θα το πιάσει η επόμενη αναδρομική κλήση.
2313	PASS	- Δεν αποδεσμεύετε τη μνήμη για κάθε μία λέξη που βρίσκεται σε κόμβο λίστας. - Η βασική περίπτωση στην αναδρομή είναι για head == NULL. Αν ο πρώτος έλεγχος είναι για το head->next, τότε αν η λίστα είναι αρχικά άδεια, θα πετάξει segfault.
2317	PASS	Δεν αποδεσμεύετε τη μνήμη για κάθε μία λέξη που βρίσκεται σε κόμβο λίστας.
2344	PASS	Δεν αποδεσμεύετε τη μνήμη για κάθε μία λέξη που βρίσκεται σε κόμβο λίστας. - Στη max_score, αν η συνθήκη της γραμμής 82 είναι αληθής, τότε επιστρέφετε τη μεταβλητή max η οποία όμως δεν έχει αρχικοποιηθεί (και περιέχει σκουπίδια).
2353	FAIL	
2354	PASS	Δεν αποδεσμεύετε τη μνήμη για κάθε μία λέξη που βρίσκεται σε κόμβο λίστας.
2362	FAIL	
2377	PASS	- Δεν αποδεσμεύεις τη μνήμη για κάθε μία λέξη που βρίσκεται σε κόμβο λίστας. - Η βασική περίπτωση στην αναδρομή είναι για head == NULL. Αν ο πρώτος έλεγχος είναι για το head->next, τότε αν η λίστα είναι αρχικά άδεια, θα πετάξει segfault.
2393	FAIL	
2418	FAIL	
2421	FAIL	
2463	PASS	