

Δομές Δεδομένων

Πάκου Ευαγγελία p18118
Κατέβας Χρήστος p18068

Ομάδα Ασκήσεων 1

Υποερώτημα α)

Struct node :

Δημιουργεί 3 ακέραιες μεταβλητές val(value), col(column), row, οι οποίες χρησιμοποιούνται στη διασυνδεδεμένη λίστα. Οι μεταβλητές αντιπροσωπεύουν αντίστοιχα την τιμή (ένος μη μηδενικού στοιχείου), την στήλη αυτού του μη μηδενικού στοιχείου και τη γραμμή αυτού (βάση της τοποθεσίας του στη μήτρα α). Δημιουργείται επίσης ένας pointer *next ο οποίος δείχνει στο επόμενο κόμβο της διασυνδεδεμένης λίστας.

Κύριο πρόγραμμα int main():

Ο χρήστης δίνει τις διαστάσεις h, w του πίνακα Ύψος : h, Πλάτος : w.

Χρήση 1 (αντί για τρεις για εξοικονόμηση χώρου) πινάκων .

Με τη βοήθεια της malloc καταλαμβάνεται ο απαιτούμενος χώρος για τη δημιουργία των μεταβλητών του struct, δηλαδή για το περιεχόμενο των “κόμβων” της διασυνδεδεμένης λίστας.

Ο τελικός πίνακας c αρχικοποιείται στο 0 (μηδέν).

Ο χρήστης εισάγει το στοιχείο που θα αντιστοιχούσε στο πίνακα και:

(Δημιουργία και επέκταση διασυνδεδεμένης λίστας)

~Αν ο αριθμός που διαβάστηκε είναι διάφορος του 0 (μηδέν)

και είναι ο πρώτος διάφορος του μηδέν τότε “δημιουργείται χώρος”(δες malloc λέγεις γραμμές πιο πάνω), ίσως για την αποθήκευση των μεταβλητών του struct node. Ο πρώτος κόμβος της διασυνδεδεμένης λίστας ονομάζεται head. Αποθηκεύονται στον κόμβο, τα απαραίτητα στοιχεία δηλαδή η μη μηδενική τιμή, η αντίστοιχη στήλη και σειρά του μη μηδενικού στοιχείου(που αντιστοιχούν στη μήτρα α) και ο pointer για τον επόμενο κόμβο γίνεται NULL (δεν ξέρουμε αν θα υπάρξει επόμενο μη μηδενικό στοιχείο.)

~Αν πάλι είναι η ν-οστή φορά που εισάγεται μη μηδενικό στοιχείο, τότε καλείται η συνάρτηση push.

Function push:

Δέχεται σαν ορίσματα τον πρώτο κόμβο μιας διασυνδεδεμένης λίστας, και τρεις ακεραίους. Διατρέχεται η λίστα μέχρι ο current (τωρινός κόμβος κάθε φορά) να στοχεύει (ο pointer) σε NULL, δηλαδή να έχουμε φτάσει στον κόμβο ο οποίος αντιπροσωπεύει το τελευταίο μη μηδενικό στοιχείο που δημιουργήθηκε.

Γίνεται δέσμευση χώρου για το περιεχόμενο του struct (με τη malloc) καταχωρούνται οι αντίστοιχες τιμές και ο επόμενος pointer στοχεύει ξανά σε NULL.

(* η στήλη και η σειρά που θα αντιστοιχούσαν στον πίνακα δίνονται από τις 2 for loop)

Γίνεται ακριβώς η ίδια διαδικασία για την εισαγωγή του πίνακα β.

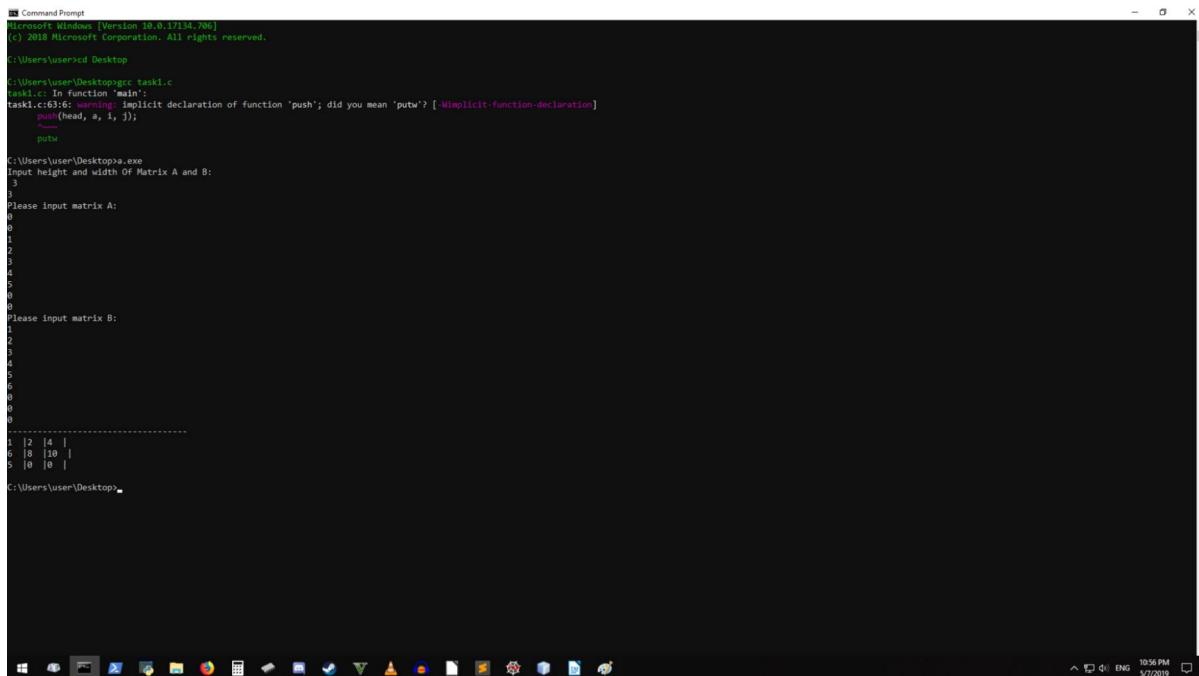
Ορίζονται τρεις μεταβλητές οι οποίες περιέχουν αντίστοιχα τη στήλη, τη σειρά και το περιεχόμενο col, row, val.

Διατρέχουμε με while loop τη διασυνδεδεμένη λίστα που αντιστοιχεί στο πίνακα α(με πρώτο κόμβο το head) εισάγωντας στον πίνακα c τα μη μηδενικά στοιχεία βάσης της στήλης και σειράς.

Επαναλαμβάνουμε τη διαδικασία για τον τη διασυνδεδεμένη λίστα του πίνακα β (με πρώτο κόμβο head2).

Τέλος εκτυπώνεται ο πίνακας c!

Screenshot πό τη εκτέλεση του προγράμματος



```
Microsoft Windows [Version 10.0.17134.700]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\user>cd Desktop
C:\Users\user\Desktop>task1.c
task1.c: In function 'main':
task1.c:63:6: warning: implicit declaration of function 'push'; did you mean 'putw'? [-Wimplicit-function-declaration]
   push(head, a, i, j);
   ~~~~
   putw

C:\Users\user\Desktop>exe
Input height and width Of Matrix A and B:
3
3
Please input matrix A:
0
0
1
2
3
4
5
6
0
Please input matrix B:
1
2
3
4
5
6
0
0
0
-----
1 |2 |4 |
6 |0 |10 |
5 |0 |0 |
C:\Users\user\Desktop>
```

Υποερώτημα β)

Struct node :

Δημιουργεί 3 ακέραιες μεταβλητές val(value), col(column), row, οι οποίες χρησιμοποιούνται στη διασυνδεδεμένη λίστα. Οι μεταβλητές αντιπροσωπεύουν αντίστοιχα την τιμή (ένος μη μηδενικού στοιχείου), την στήλη αυτού του μη μηδενικού στοιχείου και τη γραμμή αυτού (βάση της τοποθεσίας του στη μήτρα α).

Δημιουργείται επίσης ένας pointer *next ο οποίος δείχνει στο επόμενο κόμβο της διασυνδεδεμένης λίστας.

Κύριο πρόγραμμα int main():

Ο χρήστης δίνει τις διαστάσεις h, w της μήτρας a, Ύψος : h, Πλάτος : w.

Δημιουργείται μήτρα a[h][w].

Δημιουργείται μήτρα s[2][2] (διαστάσεων 2*2), η οποία θα περιέχει την πιθανή υπομήτρα και αρχικοποιείται στο 0.

Η μήτρα a γεμίζει με ψευδοτυχαίους αριθμούς από 0-9.

(Δημιουργία και επέκταση διασυνδεδεμένης λίστας)

~Αν ο ψευδοτυχαίος αριθμός που δημιουργήθηκε είναι διάφορος του 0 (μηδέν) και είναι ο πρώτος διάφορος του μηδέν τότε “δημιουργείται χώρος”, ίσος για την αποθήκευση των μεταβλητών του struct node. Ο πρώτος κόμβος της διασυνδεδεμένης λίστας ονομάζεται head. Αποθηκεύονται στον κόμβο, τα απαραίτητα στοιχεία δηλαδή η μη μηδενική τιμή, η αντίστοιχη στήλη και σειρά του μη μηδενικού στοιχείου(που αντιστοιχούν στη μήτρα a) και ο pointer για τον επόμενο κόμβο γίνεται NULL (δεν ξέρουμε αν θα υπάρξει επόμενο μη μηδενικό στοιχείο.)

~Αν πάλι είναι η n -οστή φορά που δημιουργείται μη μηδενικό στοιχείο, τότε καλείται η συνάρτηση push.

Function push:

Δέχεται σαν ορίσματα το πρώτο κόμβο μιας διασυνδεδεμένης λίστας, και τρεις ακεραίους. Διατρέχεται η λίστα μέχρι ο current (τωρινός κόμβος) να στοχεύει σε NULL, δηλαδή να έχουμε φτάσει στον κόμβο ο οποίος αντιπροσωπεύει το τελευταίο μη μηδενικό στοιχείο που δημιουργήθηκε.

Γίνεται δέσμευση χώρου για το περιεχόμενο του struct καταχωρούνται οι αντίστοιχες τιμές και ο επόμενος pointer στοχεύει ξανά σε NULL.

Τυπώνεται όλη η μήτρα a .

Ο χρήστης καλείται να δώσει έναν ακέραιο, διαβάζεται ο ακέραιος c .

Ορίζονται 3 μεταβλητές τύπου int val, col, row.

Με ακριβώς ίδιο τρόπο όπως στη συνάρτηση push διατρέχεται η διασυνδεδεμένη λίστα ξεκινώντας από το πρώτο κόμβο, την κεφαλή head.

Στις μεταβλητές val, col, row τοποθετούνται οι αντίστοιχες μεταβλητές του τρέχον κόμβου val, col, row.

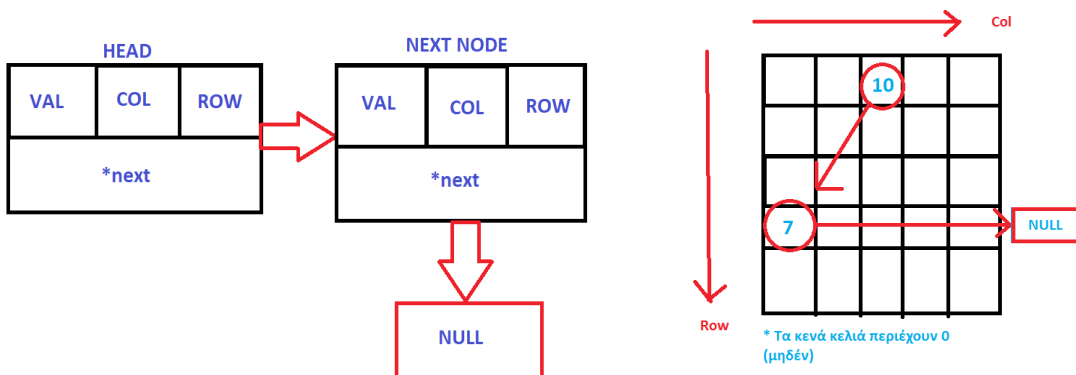
(Ακολουθούν πολλαπλές if else για την εύρεση μίας υπομήτρας της αρχικής a , της οποίας το άθροισμα των στοιχείων είναι μεγαλύτερο από τον ακέραιο c .)

Βρισκόμενοι σε δισδιάσταση μήτρα διακρίνουμε τις εξής περιπτώσεις για το που βρίσκεται το μη μηδενικό στοιχείο , του αντίστοιχου κόμβου, στη μήτρα a

- Βρίσκεται στην πρώτη γραμμή (row)
στο πρώτο κελί (χρώμα μπλε)
στο τελευταίο κελί (χρώμα ροζ)
οπουδήποτε αλλού στην πρώτη γραμμή (χρώμα ροζ και μπλε)
- Βρίσκεται στην τελευταία γραμμή (row)
στο πρώτο κελί (χρώμα πράσινο)
στο τελευταίο κελί (χρώμα πορτοκαλί)
οπουδήποτε αλλού στην τελευταία γραμμή (χρώμα πορτοκαλί και πράσινο)

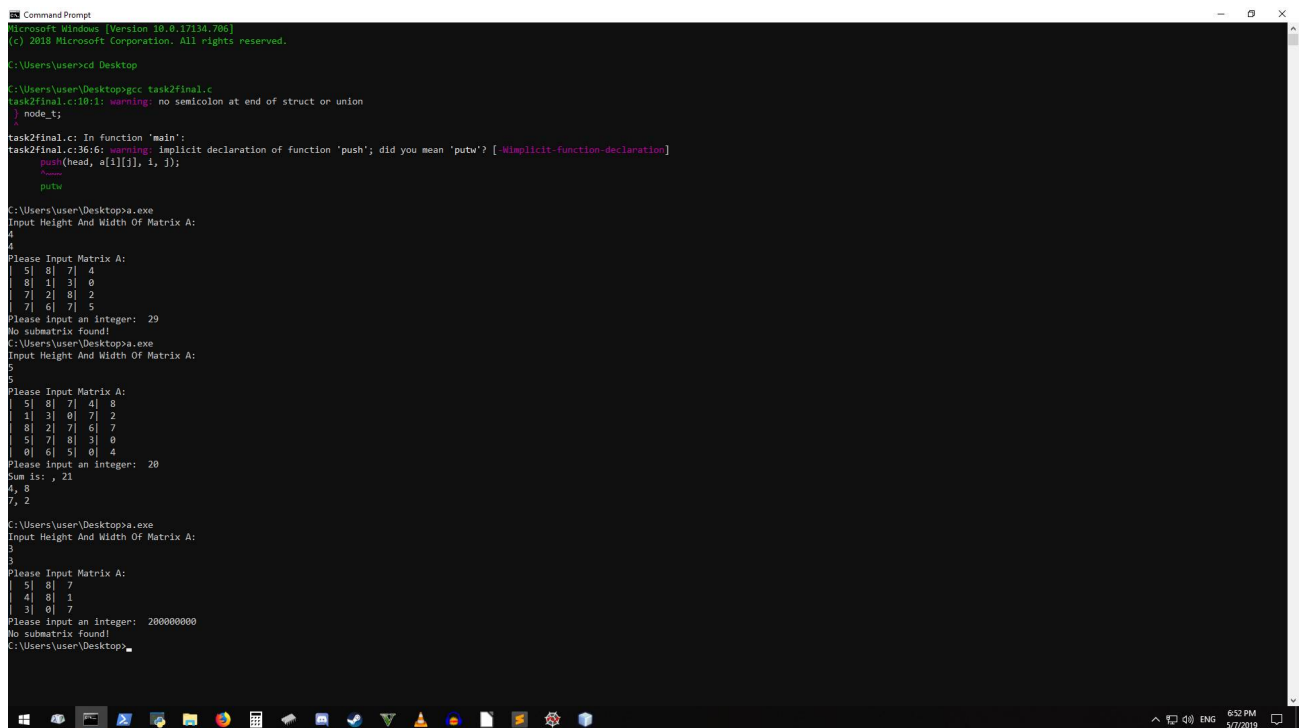
-
- The diagram shows a 3x3 grid of cells. The central cell is labeled 'COL ROW'. The cells are arranged as follows:
- | | | |
|----------------|--------------|----------------|
| COL-1
ROW-1 | COL
ROW-1 | COL+1
ROW-1 |
| COL-1
ROW | COL
ROW | COL+1
ROW |
| COL-1
ROW+1 | COL
ROW+1 | COL+1
ROW+1 |
- Red arrows point from the central cell to its eight neighbors. A green box highlights the top row of cells (COL-1 ROW-1, COL ROW-1, COL+1 ROW-1). An orange box highlights the left column of cells (COL-1 ROW, COL ROW, COL-1 ROW+1). A pink box highlights the bottom row of cells (COL-1 ROW+1, COL ROW+1, COL+1 ROW+1). A blue line highlights the right column of cells (COL+1 ROW, COL+1 ROW+1).

Ο τύπος κόμβου που χρησιμοποιήθηκε(με τη βοήθεια του struct):



Γίνεται πρόσθεση του περιεχομένου των αντίστοιχων κελιών σε μία μεταβλητή `sum`, τύπου ακεραίου, τοποθετούνται στην υποψήφια υπομήτρα `s` τα αντίστοιχα κελία. Αν το άθροισμα `sum` είναι μεγαλύτερο του `c` τότε παραλήγονται οι υπόλοιπες περιπτώσεις, τυπώνεται το άθροισμα και η υπομήτρα `s`. Σε άλλη περίπτωση εμφανίζεται το αντίστοιχο μήνυμα.

Screenshot πό τη εκτέλεση του προγράμματος



```
Microsoft Windows [Version 10.0.17134.706]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\user>cd Desktop

C:\Users\user\Desktop>gcc task2final.c
task2final.c:10:1: warning: no semicolon at end of struct or union
} node_t;
^
task2final.c: In function 'main':
task2final.c:36:6: warning: implicit declaration of function 'push'; did you mean 'putw'? [-Wimplicit-function-declaration]
    push(head, a[i][j], i, j);
    ^~~~~
task2final.c:36:6: note: 'push' was declared here
    push(head, a[i][j], i, j);
    ^~~~~

C:\Users\user\Desktop>.exe
Input Height And Width Of Matrix A:
4
Please Input Matrix A:
5| 8| 7| 4
8| 1| 3| 0
7| 2| 8| 2
7| 6| 7| 5
Please Input an Integer: 29
No submatrix found!

C:\Users\user\Desktop>.exe
Input Height And Width Of Matrix A:
5
Please Input Matrix A:
5| 8| 7| 4| 8
1| 3| 0| 7| 2
8| 2| 7| 6| 7
5| 7| 8| 3| 0
0| 6| 5| 0| 4
Please Input an Integer: 20
Sum is: 21
4, 8
7, 2

C:\Users\user\Desktop>.exe
Input Height And Width Of Matrix A:
3
Please Input Matrix A:
5| 8| 7
4| 8| 1
3| 0| 7
Please Input an Integer: 20000000
No submatrix found!

C:\Users\user\Desktop>
```

Πολυπλοκότητα:

Η χειρότερη περίπτωση είναι να είναι όλα τα κελιά της μήτρας με μη μηδενικά στοιχεία, αλλά το άθροισμα που θα ξεπερνά τον αριθμό που δίνεται να είναι στην τελευταία τετράδα!

Σε αυτή τη περίπτωση το πρόγραμμα θα εκτελέσει όλες τις `if` και μόνο στη τελευταία περίπτωση της τελευταίας λούπας θα βγάλει αποτέλεσμα!

Beta Version υποερώτημα 1)

(Προσπάθεια για ελάχιστη κατανάλωση χώρου στη κύρια μνήμη)

Struct node :

Δημιουργεί 3 ακέραιες μεταβλητές `val(value)`, `col(column)`, `row`, οι οποίες χρησιμοποιούνται στη διασυνδεδεμένη λίστα. Οι μεταβλητές αντιπροσωπεύουν

αντίστοιχα την τιμή (ένος μη μηδενικού στοιχείου), την στήλη αυτού του μη μηδενικού στοιχείου και τη γραμμή αυτού (βάση της τοποθεσίας του στη μήτρα α). Δημιουργείται επίσης ένας pointer *next ο οποίος δείχνει στο επόμενο κόμβο της διασυνδεδεμένης λίστας.

Κύριο πρόγραμμα int main():

Εισάγεται από το χρήστη το ύψος και το πλάτος της μήτρας.

Δημιουργούνται και προσθέτονται δύο ψευδοτυχαίες τιμές:

~Σκοπός είναι να προσθέσουμε τις μη μηδενικές τιμές της μήτρας α και β .

Άρα δημιουργούμε μία και μία τυχαία τιμή αντίστοιχα για κάθε πίνακα και τις προσθέτουμε κατευθείαν

- Στη περίπτωση που είναι 0 και τα 2 θα αγνοηθούν.
- Στη περίπτωση που ένα από τα δύο είναι 0 θα αγνοηθεί το αντίστοιχο αυτό μηδενικό στοιχείο.

Όπως και έχει περιγραφεί και στα προηγούμενα υποερωτήματα:

Η δημιουργία τυχαίων αριθμών περιορίζεται στο 0 και 1 για να είναι μεγαλύτερες οι πιθανότητες να δημιουργηθούν μηδενικά στοιχεία!

~Αν ο ψευδοτυχαίος αριθμός που δημιουργήθηκε είναι διάφορος του 0 (μηδέν) και είναι ο πρώτος διάφορος του μηδέν τότε “δημιουργείται χώρος”, ίσος για την αποθήκευση των μεταβλητών του struct node. Ο πρώτος κόμβος της διασυνδεδεμένης λίστας ονομάζεται head. Αποθηκεύονται στον κόμβο, τα απαραίτητα στοιχεία δηλαδή η μη μηδενική τιμή, η αντίστοιχη στήλη και σειρά του μη μηδενικού στοιχείου(που αντιστοιχούν στη μήτρα α) και ο pointer για τον επόμενο κόμβο γίνεται NULL (δεν ξέρουμε αν θα υπάρξει επόμενο μη μηδενικό στοιχείο.)

~Αν πάλι είναι η ν-οστή φορά που δημιουργείται μη μηδενικό στοιχείο, τότε καλείται η συνάρτηση push.

Function push:

Δέχεται σαν ορίσματα το πρώτο κόμβο μιας διασυνδεδεμένης λίστας, και τρεις ακεραίους. Διατρέχεται η λίστα μέχρι ο current (τωρινός κόμβος) να στοχεύει σε NULL, δηλαδή να έχουμε φτάσει στον κόμβο ο οποίος αντιπροσωπεύει το τελευταίο μη μηδενικό στοιχείο που δημιουργήθηκε.

Γίνεται δέσμευση χώρου για το περιεχόμενο του struct καταχωρούνται οι αντίστοιχες τιμές και ο επόμενος pointer στοχεύει ξανά σε NULL.

Έτσι προσθέτουμε κατευθείαν τις τιμές “των μητρών α και β ” και δημιουργούμε τη διασυνδεδεμένη λίστα των μη μηδενικών στοιχείων χωρίς όμως να καταναλώνουμε χώρο για να τις αποθηκεύσουμε πραγματικά (τις μήτρες α, β)!

Αντίστοιχα δεν απαιτείται χώρος και για την αποθήκευση της μήτρας c , καθώς ήδη έχουμε αποθηκεύσει τα απαραίτητα στοιχεία της (τα μη μεδινκά στοιχεία και τις συντεταγμένες αυτών (οι αντίστοιχη στήλη και σειρά που θα βρίσκονταν στη μήτρα c))στη διασυνδεδεμένη λίστα.

Τέλος εκτελώντας 2 εμφωλευμένες for loop:

Ξεκινώντας από την κεφαλή κόμβο της διασυνδεδεμένης λίστας, σε περίπτωση που οι 2 τελεστές των for loop αντιστοιχούν στις συντεταγμένες του μη μηδενικού στοιχείου, που έχουν αποθηκευτεί στη διασυνδεδεμένη λίστα, ΤΥΠΩΝΟΥΜΕ το μη μηδενικό στοιχείο. Και προχωράμε στον επόμενο κόμβο!

Αλλιώς τυπώνουμε 0.

Χώρος που απαιτείται:

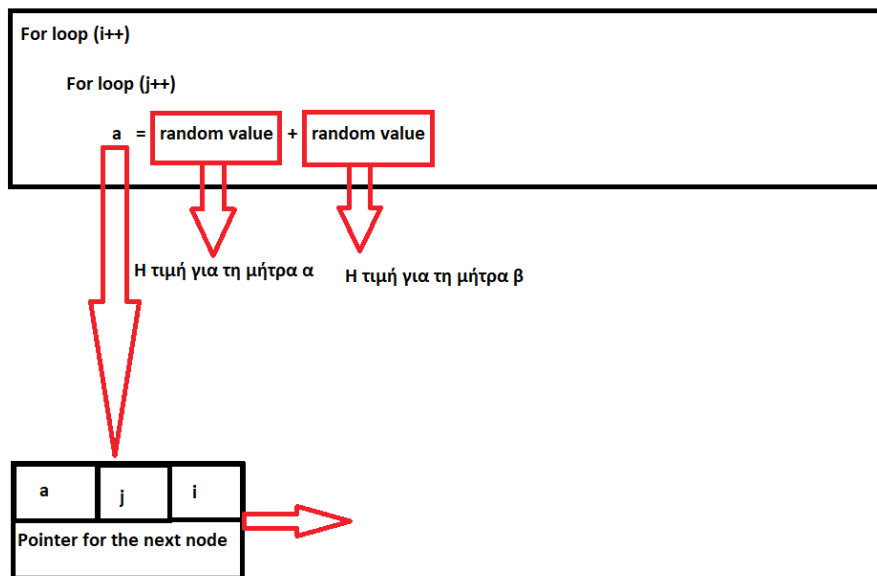
(Beta Version)

Τα στοιχεία της διασυνδεδεμένης λίστας και 6 μεταβλητές ακαίρεου τύπου.

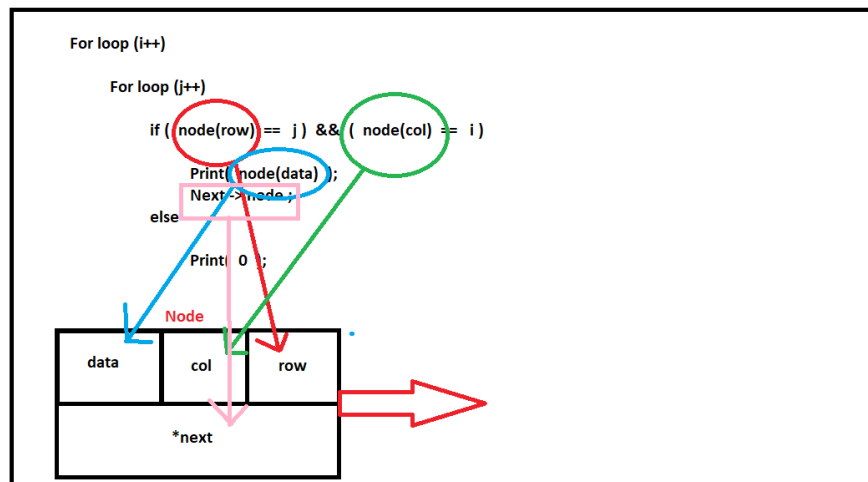
(Κανονικά)

1 πίνακας

τα στοιχεία 2 διασυνδεδεμένων λιστών και μεταβλητές.



Πώς λειτουργεί η εκτύπωση συνολικά της μήτρας-άθροισμα μόνο με τη διασυνδεδεμένη λίστα



Screenshot πό τη εκτέλεση του προγράμματος

```
Command Prompt
Microsoft Windows [Version 10.0.17134.706]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\User>cd Desktop

C:\Users\User\Desktop>gcc beta_task1.c
beta_task1.c: In function 'main':
beta_task1.c:22:6: warning: implicit declaration of function 'push'; did you mean 'putw'? [-Wimplicit-function-declaration]
     push(head, a, i, j);
     ^~~~~
     putw
beta_task1.c:46:2: warning: implicit declaration of function 'print_list'; did you mean 'printf_s'? [-Wimplicit-function-declaration]
     print_list(head, h, w);
     ^~~~~~
     printf_s

C:\Users\User\Desktop>pa.exe
Input height and width of matrix a:
5
5
0 | 0 | 1 | 0 | 0 |
2 | 2 | 2 | 1 | 1 | 1 |
1 | 1 | 0 | 1 | 1 | 1 |
1 | 1 | 1 | 2 | 1 | 1 |
2 | 1 | 1 | 2 | 1 | 1 |

C:\Users\User\Desktop>
```