



Εξέταση Φεβρουαρίου 2023

Ερώτημα 1

- a) Με ποια εντολή μπορείτε να δείτε σε ποιο φάκελο βρίσκεστε;
- b) Πώς μπορείτε να εμφανίσετε μια λίστα με όλα τα αρχεία, συμπεριλαμβανομένων των κρυφών αρχείων;
- c) Η εντολή Fork() χρησιμοποιείται για τη δημιουργία μιας νέας διεργασίας;
- d) Από τον ακόλουθο κώδικα σε C θα πρέπει να κάνετε τις αναγκαίες αλλαγές για να εμφανίσει ΜΟΝΟ το λειτουργικό που τρέχει το πρόγραμμα, ποιος είστε (το username σας) και τη πλήρη ημερομηνία κατά την εκτέλεση του προγράμματος. Η απάντηση πρέπει να είναι με χρήση εντολών Linux και όχι με echo, δηλαδή το πρόγραμμα πρέπει να είναι δυναμικό.

Το αρχείο μπορείτε να το βρείτε και στο φάκελο /exams2023/

```
#include <stdio.h>
#include <stdlib.h>
```

```
#define ENTERCODEHERE "\
#/bin/bash \n\
echo \"Anti gia auti ti seira tha exete oses xreistoun gia na valetе tis entoles sas\" \n\
"
```

```
int main()
{
    puts("Tha treksei to script pou eftiaksa kai perixeі ta akoloutha :");
    puts(ENTERCODEHERE);
    puts("To epotelesma tou einai :");
    system(ENTERCODEHERE);
    return 0;
}
```

Ερώτημα 2

Έχοντας ένα αρχείο που περιέχει τα εξής:

```
name;year;mark;times
Vrettos;3;3;1
Lydia;7;10;1
Eva;1;10;2
Tasos;2;5;3
Kostas;1;6;4
Maria;3;2;2
Dimitra;6;5;3
Aliko;5;1;4
George;2;9;2
Dimitris;7;7;1
Georgia;4;3;1
Patrick;2;4;2
Filoktitis;3;0;6
Petros;8;6;3
Eliza;4;8;2
Vasilis;5;5;4
Stamatis;2;8;1
Lefteris;2;7;2
Pavlina;4;3;4
Theovrastos;6;4;1
Despoina;3;2;3
Patrikios;2;6;8
Stamatina;4;8;4
Lina;7;4;4
```

Το αρχείο είναι και στο /exams2023 με όνομα sample.txt

Που περιέχει το όνομα του φοιτητή τη χρονιά που είναι, το βαθμό που πήρε στην εξεταστική και τις φορές που έχει δώσει το μάθημα ζητείται

A)Το υπουργείο αποφάσισε ότι όσοι παλαιοί φοιτητές έχουν γράψει σε αυτή την εξεταστική πάνω ή ίσο με 3 και έχουν δώσει αρκετές φορές το μάθημα να τους δοθεί η δυνατότητα να περάσουν το μάθημα.

Οπότε αλγοριθμικά (με AWK) θα πρέπει να επιστραφούν οι φοιτητές που είναι μεγαλύτερο από 5 έτος ΚΑΙ έχουν πάρει βαθμό μεγαλύτερο ή ίσο με 3 ΚΑΙ μικρότερο από 5 στην εξέταση ΚΑΙ έχουν δώσει το μάθημα πάνω από 2 φορές.

Η απάντηση εδώ θα πρέπει να είναι δηλαδή

```
Lina;7;4;4
```

B) Στον ακόλουθο κώδικα επιστρέφεται ο μέσος όρος και ο συνολικός αριθμός των ατόμων για το παράδειγμα από πάνω

```
awk -F';' 'NR>=2 {{sum+= $3;count+=1;}} END{print sum/count; print count;}' sample.txt
```

5.25

24

Πως πρέπει να τροποποιηθεί ο κώδικας ώστε εάν το τρέχουμε και άδειο αρχείο να μην μας επιστρέφει αυτό το μήνυμα; (το αρχείο sample_empty.txt είναι στο φάκελο /exams2023)

```
awk -F',' 'NR>=2 {{sum+=$3;count+=1;}} END{print sum/count; print count;}' sample_empty.txt  
awk: cmd. line:1: (FILENAME=sample_empty.txt FNR=1) fatal: division by zero attempted
```

Ερώτημα 3

Έχοντας μια βάση με φοιτητές που έχουν τους βαθμούς τους θα χρειαστεί να τροποποιήσετε το κώδικα ώστε να επιστρέφει το μέσο όρο των βαθμολογιών από τη βαθμολογία που είναι μεγαλύτερη ή ίδια από αυτή που θα δίνει ο χρήστης. Δηλαδή εάν δώσει πάνω από

```
# ./a.out mech.db '6'  
posoi einai: 4  
to athroisma: 30  
o meros oros: 7.500000
```

```
# ./a.out mech.db '9'  
posoi einai: 1  
to athroisma: 9  
o meros oros: 9.000000
```

```
# ./a.out mech.db '1'  
posoi einai: 8  
to athroisma: 39  
o meros oros: 4.875000
```

Οι εντολές για να φτιαχτούν οι πίνακες
sqlite3 mech.db < makedb.sql
sqlite3 mech.db < insert.sql

makedb.sql

```
CREATE TABLE IF NOT EXISTS programming(name,mark);
```

insert.sql

```
INSERT INTO programming(name,mark) VALUES ('Vrettos Moulos','4');  
INSERT INTO programming(name,mark) VALUES ('Pavlos Kranas','2');  
INSERT INTO programming(name,mark) VALUES ('Anna Gantzioura','7');  
INSERT INTO programming(name,mark) VALUES ('Donald Knuth','9');  
INSERT INTO programming(name,mark) VALUES ('Christos  
Papadimitriou','2');  
INSERT INTO programming(name,mark) VALUES ('Linus Torvald','1');  
INSERT INTO programming(name,mark) VALUES ('Lydia Kalogeropoulou','8');  
INSERT INTO programming(name,mark) VALUES ('Eva Drogkari','6');
```

Εάν θέλετε το κώδικα μπορείτε να το πάρετε και από το /exams2023

Η εντολή για αντιγραφή από το home folder σας είναι

```
cp /exams2023/insert.sql .
```

cp /exams2023/makedb.sql .

cp /exams2023/mech.db .

ο κώδικας

```
#include <stdio.h>
// header file για το βάση δεδομένων
#include <sqlite3.h>

int main(int argv, char **argc)
{
    // Περιέχει το ερώτημα που θέλουμε να δοκιμάσουμε στη βάση
    char* sql = "SELECT name, mark FROM programming WHERE name = ?1;";
    // pointer για τη βάση δεδομένων που έχουν φτιάξει
    sqlite3 *db;
    // το statement που δημιουργούμε για το SQL ερώτημα που έχουμε
    // ορίσει πιο πάνω
    sqlite3_stmt *pstmt;
    // περιέχουν τα αποτελέσματα κάθε ερωτήματος
    const unsigned char *name,*mark;
    // αρχικοποίηση για το pointer λάθους
    char *zErrMsg = 0;
    // απο τα αρχικά return code , δηλαδή έχει την επιστροφή απο τη
    // SQLite
    int rc;

    // έλεγχος σωστών ορισμάτων
    if (argv != 3)
    {
        printf("the right way to call the software is : %s name of the
        db and the name of the student \n",argc[0]);
        return(1);
    }

    // standard command για να ανοίξει η βάση δεδομένων
    // argc[1] --> database file name , &db --> SQLite db handle ,
    // SQLITE_OPEN_READONLY --> extra flag , NULL --> SQLite OS portability
    // layer είτε όπως θα το βρείτε και στο internet ως "VFS", τ>
    rc = sqlite3_open_v2(argc[1], &db, SQLITE_OPEN_READONLY, NULL);
    // printf("%d\n\n", rc);
    if (rc)
        // if (db == NULL)
    {
        fprintf(stderr, "Can't open database: %s\n",
        sqlite3_errmsg(db));
        // Η συνάρτηση που καταστρέφει ότι έχει ανοίξει το αντικείμενο
        // (pointer) της SQLite
        sqlite3_close_v2(db);
        return(1);
    }

    // prepared statement db --> db handler , το ερώτημα SQL ,
    // Maximum length , flags , η έξοδος του Statement handle , Εάν δεν είναι
    // NULL τότε γίνεται για να δείχνει στο πρώτο byte μετά το >
    rc = sqlite3_prepare_v3(db, sql, -1, 0, &pstmt, NULL);
    if (rc)
    {
        fprintf(stderr, "Could not prepare SQL statement: %s \n",
```

```

sqlite3_errmsg(db));
    // καλείται για διαγραφή της δήλωσης
    sqlite3_finalize(pstmt);
    sqlite3_close_v2(db);
    return(1);
}

// Αλλάζει το ?1 στο ερώτημα με το text που του έχουμε δώσει
// int sqlite3_bind_text(sqlite3_stmt*,int,const
char*,int,void*)(void*);
    // 1st argument pstmt --> είναι ο pointer για το sqlite3_stmt
    που επιστρέφεται από τη sqlite3_prepare_v2()
    // 2nd argument 1 --> είναι το ευρετήριο της παραμέτρου SQL που
    πρόκειται να οριστεί. Η αριστερή παράμετρος SQL έχει ένα ευρετήριο 1.
    // Όταν η ίδια παράμετρος SQL που χρησιμοποιείται,
    χρησιμοποιείται περισσότερες από μία φορές, τα επακόλουθα συμβάντα
    έχουν τον ίδιο δείκτη με την πρώτη εμφάνιση.
    // 3rd argument argc[2] --> είναι η τιμή που συνδέεται με την
    παράμετρο
    // 4th argument -1 --> το μήκος της συμβολοσειράς είναι ο
    αριθμός των byte έως τον πρώτο μηδενικό τερματιστή
    rc = sqlite3_bind_text(pstmt, 1, argc[2], -1, NULL);
    if (rc)
    {
        fprintf(stderr, "Couldn't bind to prepared sql stmt: %s\n",
sqlite3_errmsg(db));
        sqlite3_finalize(pstmt);
        sqlite3_close_v2(db);
        // non-zero (usually 1, but not necessarily) in case of
error
        return(1);
    }
    // loop για όλες τις εγγραφές
    while(sqlite3_step(pstmt) == SQLITE_ROW)
    {
        name = sqlite3_column_text(pstmt,0);
        mark = sqlite3_column_text(pstmt,1);
        printf("name: %s mark: %s \n",name,mark);
    }
    // διαγραφή της δήλωσης
    sqlite3_finalize(pstmt);

    // καταστροφή του pointer
    sqlite3_close_v2(db);

    return 0;
}

```

Για να γίνει compile πρέπει να τρέξετε την εντολή

`gcc -O2 Exam_Question_3.c -lsqlite3`

(Αυτό βέβαια εάν έχετε ονομάσει το αρχείο Exam_Question_3.c)

Μπορείτε το αρχείο να το πάρετε από το φάκελο /exams2023 . Η

εντολή για αντιγραφή είναι το `cp /exams2023/Exam_Question_3.c` .

Από το internet σας παρουσιάζω ένα παράδειγμα με την atoi που μπορείτε να χρησιμοποιήσετε. (Μετατρέπει ένα string σε integer)

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main () {
    int val;
    char str[20];

    strcpy(str, "98993489");
    val = atoi(str);
    printf("String value = %s, Int value = %d\n", str, val);

    strcpy(str, "tutorialspoint.com");
    val = atoi(str);
    printf("String value = %s, Int value = %d\n", str, val);

    return(0);
}
```

 Live Demo

Let us compile and run the above program that will produce the following result –

```
String value = 98993489, Int value = 98993489
String value = tutorialspoint.com, Int value = 0
```

Ερώτημα 4

- a) Να συγκριθούν με το θεώρημα κυριαρχίας οι αναδρομικές συναρτήσεις και να οργανωθούν από αυτή που θέλει υπολογιστικά λιγότερο σε αυτή που θέλει περισσότερο χρόνο.

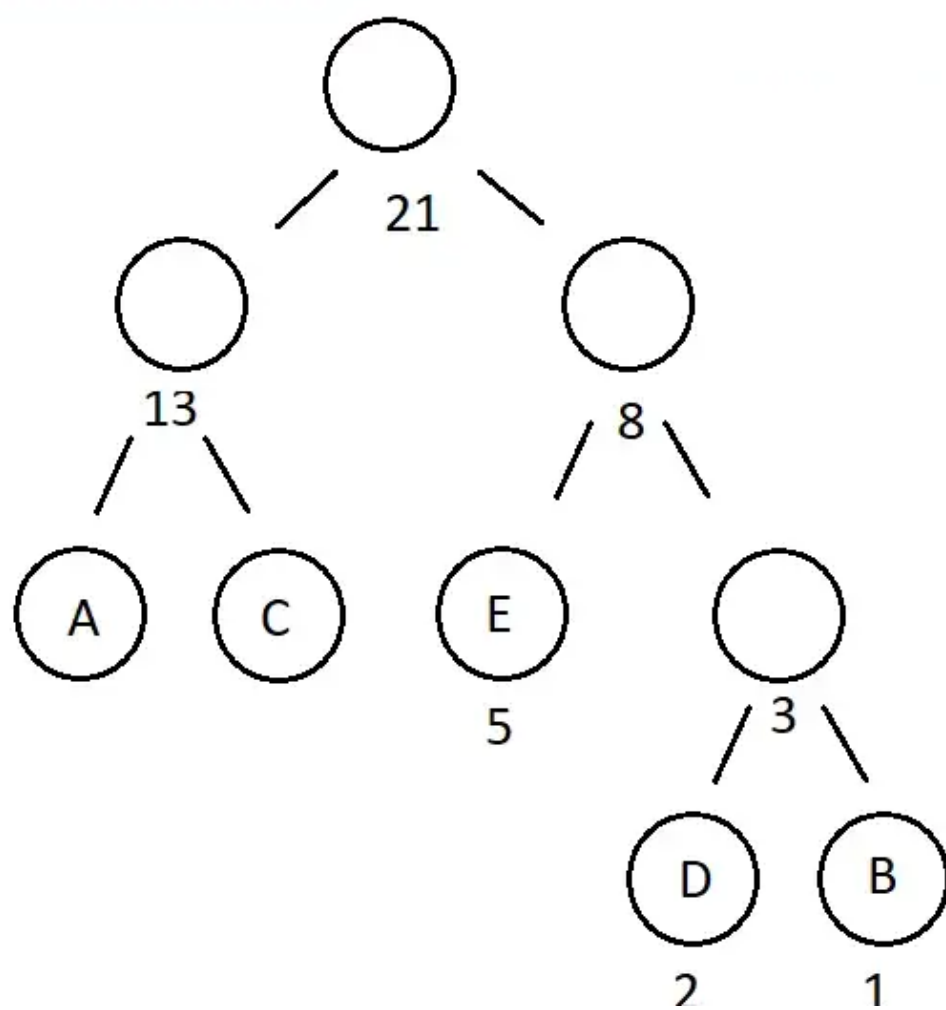
$$T(n) = 2T\left(\frac{n}{2}\right) + n$$

$$T(n) = 2T\left(\frac{n}{2}\right) + \sqrt{n}$$

$$T(n) = 3T\left(\frac{n}{4}\right) + n \lg n$$

$$T(n) = 2T\left(\frac{n}{2}\right) + n^2$$

- b) Θα μπορούσε στο ακόλουθο διάγραμμα Huffman να ήταν στο A και στο C οι αριθμοί 4 και 9; Εάν είναι ακέραιοι μόνο οι αριθμοί τι αριθμούς θα βάζατε εάν είναι βέβαια λάθος το 4 και το 9;



Ερώτημα 5

a)

Τι πρόβλημα μπορεί να έχει το ακόλουθο εκτελέσιμο;

```
valgrind ./a.out
==1750803== Memcheck, a memory error detector
==1750803== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et
al.
==1750803== Using Valgrind-3.16.1 and LibVEX; rerun with -h for
copyright info
==1750803== Command: ./a.out
==1750803==
==1750803== Invalid write of size 1
==1750803==    at 0x109171: main (in /home/debian/a.out)
==1750803==    Address 0x4a2704a is 0 bytes after a block of size 10
alloc'd
==1750803==    at 0x483877F: malloc (vg_replace_malloc.c:307)
==1750803==    by 0x109156: main (in /home/debian/a.out)
==1750803==
==1750803== HEAP SUMMARY:
==1750803==    in use at exit: 0 bytes in 0 blocks
==1750803==    total heap usage: 1 allocs, 1 frees, 10 bytes allocated
==1750803==
==1750803== All heap blocks were freed -- no leaks are possible
==1750803==
==1750803== For lists of detected and suppressed errors, rerun with: -s
==1750803== ERROR SUMMARY: 5 errors from 1 contexts (suppressed: 0 from
0)
```

b)

Τι πρόβλημα μπορεί να έχει το ακόλουθο εκτελέσιμο;

```
==1750865== Memcheck, a memory error detector
==1750865== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et
al.
==1750865== Using Valgrind-3.16.1 and LibVEX; rerun with -h for
copyright info
==1750865== Command: ./a.out
==1750865==
==1750865== Invalid free() / delete / delete[] / realloc()
==1750865==    at 0x48399AB: free (vg_replace_malloc.c:538)
==1750865==    by 0x109172: main (in /home/debian/a.out)
==1750865==    Address 0x4a27040 is 0 bytes inside a block of size 54
free'd
==1750865==    at 0x48399AB: free (vg_replace_malloc.c:538)
==1750865==    by 0x109166: main (in /home/debian/a.out)
==1750865==    Block was alloc'd at
==1750865==    at 0x483877F: malloc (vg_replace_malloc.c:307)
==1750865==    by 0x109156: main (in /home/debian/a.out)
==1750865==
==1750865== HEAP SUMMARY:
==1750865==    in use at exit: 0 bytes in 0 blocks
==1750865==    total heap usage: 1 allocs, 2 frees, 54 bytes allocated
==1750865==
==1750865== All heap blocks were freed -- no leaks are possible
```

```
==1750865==  
==1750865== For lists of detected and suppressed errors, rerun with: -s  
==1750865== ERROR SUMMARY: 1 errors from 1 contexts (suppressed: 0 from 0)
```

c)

Τι πρόβλημα μπορεί να έχει ο ακόλουθος κώδικας;

```
1 #include <stdio.h>  
2 #include <stdlib.h>  
3  
4  
5  
6 int main()  
7 {  
8     int* p1, p2;  
9  
10    int n = 30;  
11  
12    p1 = &n;  
13  
14    p2 = &n;  
15 }  
16
```

d)

Τι πρόβλημα μπορεί να έχει ο ακόλουθος κώδικας;

```
1 #include <stdio.h>  
2 #include <stdlib.h>  
3  
4  
5 int main()  
6 {  
7     int* p1; // create a pointer to an integer  
8     int m = 100;  
9     p1 = &m; // assign address of m to p1  
10  
11     *p1++; // ERROR: we did not increment value of m  
12  
13     printf("%d\n", *p1);  
14     printf("%d\n", m);  
15  
16     return 0;  
17 }
```

Συμπληρωματικές πληροφορίες

Για τη σύνδεση στο server πρέπει να χρησιμοποιήσετε το kitty/putty με στοιχεία Server mech.grid.ece.ntua.gr είτε την IP του 62.217.82.167

Μπορείτε να χρησιμοποιήσετε το man από το Linux και τις σημειώσεις σας. Δεν μπορείτε να χρησιμοποιήσετε το internet (όπου είναι και αιτία μηδενισμού)

Username είναι η θέση σας. Δηλαδή εάν είστε στο έδρανο 9 τότε είστε το student9 και ο κωδικός σας είναι ο ίδιος.