Datum: Sun, 26 Dec 2010 21:46:22 +0100 [2010-12-26 21:46:22 CET]
Od: Vrabel Lukas <ivrabel@fit.vutbr.cz>
Komu: xloffa00@stud.fit.vutbr.cz
Předmět: IZP - hodnocení projektu 3
Vytištěno: Loffay Pavol

```
--------------------------------------------------------------------------------
IZP - hodnocení projektu 3 -> xloffa00@stud.fit.vutbr.cz
--------------------------------------------------------------------------------
Reklamace posílejte na adresu opravujícího: ivrabel@fit.vutbr.cz

xloffa00-fit: celkem 9.5b

Bližší vysvětlení některých chyb najdete v publikaci "Nedělejte zbytečné chyby
v jazyce C" (http://www.fit.vutbr.cz/~martinek/clang/noerrors.html). Například
označení "5.5 Indexace za hranicí pole" znamená číslo a název kapitoly
s popisem tohoto problému. Nejprve se prosím do této publikace podívejte
a až potom případně konzultujte svůj výsledek odpovědí na tento email.

Poznámky:
ad obhajoba:

ad překlad:


ad funkčnost:
  --test:
  --vadd:
  --vscal:
  --mmult:
  --mexpr: částečně -0.5b

  --eight:
  --bubbles:
  --maze:
  --carom:
  OK

  poškozený vstup a chybné parametry: OK

ad implementace:


ad překlad:
xloffa00-fit
gcc -std=c99 -pedantic -Wall -W -g -O -o proj3 proj3.c
*** PŘEKLAD OK

Opravil Lukas Vrabel, ivrabel@fit.vutbr.cz

######### Protokol o překladu + výsledky automatických testů #########
--------------- Statická analýza kódu ----------------------
Analýza kódu:
xloffa00-fit
Testy nad proj3.c:
  **********
  *** proj3.c: Hledám goto
  *** OK, není tu
  **********
  *** proj3.c: Hledám scanf.*%s
  *** OK, není tu
  **********
  *** proj3.c: Test fopen -> fclose
354:    if ((fr = fopen(fileName, "r")) == NULL)
407:    if (( fr = fopen(fileName, "r")) == NULL)
469:    if (( fr = fopen(fileName, "r")) == NULL)
592:    if ((fr = fopen(fileName, "r")) == NULL)
```

```
362:        fclose(fr);
368:        fclose(fr);
373:        fclose(fr);
381:            fclose(fr);
389:        fclose(fr);
392:    fclose(fr);
415:        fclose(fr);
421:        fclose(fr);
427:        fclose(fr);
433:        fclose(fr);
443:                fclose(fr);
451:        fclose(fr);
454:    fclose(fr);
478:        fclose(fr);
485:        fclose(fr);
492:        fclose(fr);
499:        fclose(fr);
505:        fclose(fr);
517:                    fclose(fr);
526:        fclose(fr);
529:    fclose(fr);
601:        fclose(fr);
605:    fclose(fr);
*** OK
  **********
  *** proj3.c: Test malloc\|calloc\|realloc -> free
18:// pre free a malloc
180:                result.fileNameA = (char*) malloc(length * sizeof(char));
196:                result.fileNameA = (char*) malloc(length * sizeof(char));
204:                    result.fileNameB = (char*) malloc(length * sizeof(char));
228:    vector->array = (int *)malloc(vector->columns * sizeof(int));
244:    matrix->array = malloc(matrix->rows * sizeof(int*));  //riadky
252:        matrix->array[i] = (int *)malloc(matrix->columns * sizeof(int));
272:    matrix->array = (int ***)malloc(matrix->number * sizeof(int**)); //matice
281:        matrix->array[i] = (int **)malloc(matrix->rows * sizeof(int*));
//alokujem pocet riadkov v kazdej matici
296:            matrix->array[i][j] = (int *)malloc(matrix->columns * sizeof(int));
1199:    index = (int *)malloc(matrix.columns * sizeof(int));
1207:    identificator = (int *)malloc(matrix.columns * sizeof(int));
18:// pre free a malloc
207:                        free(result.fileNameA);
256:                free(matrix->array[j]);
257:            free(matrix->array); //uvolnim celu
285:                free(matrix->array[x]);
286:            free(matrix->array);
300:                    free(matrix->array[i][k]);
304:                        free(matrix->array[x][y]);  //dealokujem ostatne riadky
306:                free(matrix->array);
319:void freeMatrix(TMatrix *matrix)
322:        free(matrix->array[i]);
323:    free(matrix->array);
330:void freeVectorMatrix(TVectorMatrix *matrix)
335:            free(matrix->array[i][j]);
338:        free(matrix->array[i]);
339:    free(matrix->array); //uvolni matice
380:            free(vector->array);
388:        free(vector->array);
442:                freeMatrix(matrix);
450:        freeMatrix(matrix);
516:                    freeVectorMatrix(matrix);
525:        freeVectorMatrix(matrix);
612:            free(vector.array);
624:                freeMatrix(&matrix);
636:                    freeVectorMatrix(&matrix);
679:        free(vectorA.array);
685:        free(vectorA.array);
686:        free(vectorB.array);
695:        free(vectorA.array);
696:        free(vectorB.array);
```

```
701:    free(vectorResult.array);
702:    free(vectorA.array);
703:    free(vectorB.array);
743:        free(vectorA.array);
749:        free(vectorA.array);
750:        free(vectorB.array);
757:    free(vectorA.array);
758:    free(vectorB.array);
803:        freeMatrix(&matrixA);
809:        freeMatrix(&matrixA);
810:        freeMatrix(&matrixB);
820:        freeMatrix(&matrixA);
821:        freeMatrix(&matrixB);
826:    freeMatrix(&matrixA);
827:    freeMatrix(&matrixB);
828:    freeMatrix(&matrixResult);
854:        freeMatrix(&matrixA);
860:        freeMatrix(&matrixA);
861:        freeMatrix(&matrixB);
871:        freeMatrix(&matrixA);
872:        freeMatrix(&matrixB);
877:    freeMatrix(&matrixB);
880:        freeMatrix(&matrixA);
881:        freeMatrix(&matrixResult);
891:        freeMatrix(&matrixA);
892:        freeMatrix(&matrixResult);
898:    freeMatrix(&matrixA);
899:    freeMatrix(&matrixResult);
900:    freeMatrix(&matrixResultFinal);
1084:        free(vector.array);
1101:    freeMatrix(&matrix);
1102:    free(vector.array);
1202:        freeMatrix(&matrix);
1210:        freeMatrix(&matrix);
1211:        free(index);
1232:    freeMatrix(&matrix);
1233:    free(index);
1234:    free(identificator);
1276:        free(params.fileNameA);
1279:        free(params.fileNameB);
*** OK
--------------- Test správnosti implementace ---------------
Výstup:
xloffa00-fit:
- - - - - - - - Test spravnosti implementace - - - - - - - -
*****************************
**** Povinne operace
********************
*** --test
**********
** ./proj3 --test v3a.txt
*** OK
** ./proj3 --test vbad1.txt
*** OK
** ./proj3 --test vbad2.txt
*** OK
** ./proj3 --test mtest.txt
*** OK
** ./proj3 --test mtestbad1.txt
*** OK
** ./proj3 --test mtestbad2.txt
*** OK
** ./proj3 --test mtestbad3.txt
*** OK
** ./proj3 --test mtestbad4.txt
*** OK

********************
*** --vadd (a)+(b)
```

```
**********
** ./proj3 --vadd v1.txt v1.txt
*** OK
** ./proj3 --vadd v2.txt v2.txt
*** OK
** ./proj3 --vadd v3a.txt v3b.txt
*** OK
** ./proj3 --vadd v3b.txt v3c.txt
*** OK

********************
*** --vscal (a)*(b)
**********
** ./proj3 --vscal v1.txt v1.txt
*** OK
** ./proj3 --vscal v2.txt v2.txt
*** OK
** ./proj3 --vscal v3a.txt v3b.txt
*** OK
** ./proj3 --vscal v3b.txt v3c.txt
*** OK
** ./proj3 --vscal v3a.txt v2.txt
*** OK

********************
*** --mmult (A*B)
**********
** (4 3) * (3 2) = (4 2)
** ./proj3 --mmult m4x3.txt m3x2.txt
*** OK
**********
** (6 1) * (1 5) = (6 5)
** ./proj3 --mmult m6x1.txt m1x5.txt
*** OK
**********
** (1 6) * (6 1) = (1 1)
** ./proj3 --mmult m1x6.txt m6x1.txt
*** OK
**********
** (3 2) * (2 2) = (3 2)
** ./proj3 --mmult m3x2.txt m2x2.txt
*** OK
**********
** (1 6) * (1 6) = false
** ./proj3 --mmult m1x6.txt m1x6.txt
*** OK
**********
** (5 😎 * (6 😎 = false
** ./proj3 --mmult m5x8.txt m6x8.txt
*** OK

********************
*** --mexpr (A*B)*A
**********
** ((3 3)*(3 3)) * (3 3) = (3 3) (nulova)
** ./proj3 --mexpr m3x3a.txt m3x3b.txt
*** OK
**********
** ((3 3)*(3 3)) * (3 3) = (3 3) (jednotkova)
** ./proj3 --mexpr m3x3b.txt m3x3c.txt
*** OK
**********
** ((3 4)*(4 3)) * (3 4) = (4 3) (nahodna)
** ./proj3 --mexpr m3x4.txt m4x3.txt
3,6c3,6
< 3 4
< 43941 73548 73368 83153
< 41218 68987 68818 77998
< 19301 32086 31988 36393
```

```
---

 3 3
 43941 73548 73368
 41218 68987 68818
 19301 32086 31988

*** NESHODUJE SE SE VZOROVYM RESENIM **********
** ((3 4)*(3 2)) * (3 4) = false
** ./proj3 --mexpr m3x4.txt m3x2.txt
*** OK

********************
*** --eight
**********
** ./proj3 --eight v1.txt m5x8.txt
*** OK
** ./proj3 --eight v2.txt m3x3c.txt
*** OK
** ./proj3 --eight v3a.txt m3x3b.txt
*** OK
** ./proj3 --eight v3d.txt m6x8.txt
*** OK
** ./proj3 --eight v3e.txt m6x8.txt
*** OK

********************
*** --bubbles
**********
** ./proj3 --bubbles m10x10a.txt
*** OK
** ./proj3 --bubbles m10x10b.txt
*** OK
** ./proj3 --bubbles m10x10c.txt
*** OK
** ./proj3 --bubbles m6x8.txt
*** OK

********************
*** --maze
**********
** ./proj3 --maze maze1.txt
Chybne parametry prikazoveho riadku! Pre napovedu -h
2d1
< 1 2 3 2 8 9 5 9 2
\ No newline at end of file
*** NESHODUJE SE SE VZOROVYM RESENIM
** ./proj3 --maze maze2.txt
Chybne parametry prikazoveho riadku! Pre napovedu -h
2d1
< 7 5 5 2 5 6 8 7 8 4 0 3 1
\ No newline at end of file
*** NESHODUJE SE SE VZOROVYM RESENIM
** slepe bludiste: (muze vypsat 1 nebo false)
** ./proj3 --maze maze3.txt
Chybne parametry prikazoveho riadku! Pre napovedu -h
*** OK

********************
*** --carom
**********
** priklad ze zadani:
** ./proj3 --carom 0,3 SV 11 carom1.txt
Chybne parametry prikazoveho riadku! Pre napovedu -h
2d1
< 1 2 1 3 2 -6 1 2 1 4 3
*** NESHODUJE SE SE VZOROVYM RESENIM
** severo-jih:
** ./proj3 --carom 0,3 J 7 carom1.txt
```

```
Chybne parametry prikazoveho riadku! Pre napovedu -h
2d1
< 1 2 3 1 3 2 1
*** NESHODUJE SE SE VZOROVYM RESENIM
** sudy pocet sloupcu:
** ./proj3 --carom 0,0 JV 22 carom2.txt
Chybne parametry prikazoveho riadku! Pre napovedu -h
2d1
< 6 0 3 2 1 1 -2 -4 2 3 1 2 3 0 2 1 3 2 -6 1 2 1
*** NESHODUJE SE SE VZOROVYM RESENIM

*****************************
**** Test reakce na chyby
**********
** Poskozene matice
** (-1 10) * (-1 10)
** ./proj3 --mmult mbad1.txt mbad1.txt
V subore sa nenachadzaju spravne data.
#
**********
** (5 4), ale obsahuje jen (2 4)
** ./proj3 --mmult mbad2.txt mbad2.txt
V subore sa nenachadzaju spravne data.
#
**********
** (3 3), ale obsahuje jen (2 3)
** ./proj3 --mmult mbad3.txt mbad2.txt
Nepodarilo sa otvorit subor / Neexistuje subor.
#
**********
** prazdny soubor
** ./proj3 --test prazdny.txt
false
#
**********
** (10000000 10000000) test nedostatku pameti
./proj3 --test mbig.txt
false
#
**********
** vadny parametr
** ./proj3 -ble m2x2.txt
Chybne parametry prikazoveho riadku! Pre napovedu -h
#
**********
** chybi soubor
** ./proj3 --add m2x2.txt
Chybne parametry prikazoveho riadku! Pre napovedu -h
#
**********
** Napoveda (neni autorem Kacer Donald?)
Program Maticove operacie
Autor: Pavol Loffay 1BIT kruzok 30.
Pouzitie, v prikazovom riadku:
$ ./proj3 -h                  :program vypise tuto napovedu
$ ./proj3 --test a.txt        :program otestuje spravny format vstupneho suboru
$ ./proj3 --vadd a.txt b.txt  :program pocita sucet vektorov (a+b)
$ ./proj3 --vscal a.txt b.txt    :program pocita skalarny sucin vektorov (a*b)
$ ./proj3 --mmult a.txt b.txt    :program pocita sucin dvoch matic (A*B)
$ ./proj3 --mexpr a.txt b.txt    :program pocita vyraz (A*B*A)
$ ./proj3 --eight a.txt b.txt    :program hlada vektor z a.txt v matici b.txt
$ ./proj3 --bubbles a.txt    :program pocita pocet bublin v matici a.txt
#

--------------------------------------------------------------------------------
```