

| | |
|-------------------|-------|
| מחברת מס' _____ | 76634 |
| מתוך _____ מחברות | |



אוניברסיטת תל-אביב

213

**לפני התחלת הבחינה מלא את
וקרא בעיון**

תאריך הבחינה 22/6/03
שם הקורס 1 אינץ'
שם המורה 3 ברק 3 זן האפרין
החוב/המגמה 8? א' ה נחש

הוראות

1. על הנבחן להיבחן רק בחדר שבו הוא רשום.
2. עם הכניסה לחדר הבחינה יש להניח את החפצים בצד לרבות מכשירי קשר ואמצעי תקשורת אחרים כשהם כבויים.
3. אסור להחזיק בהישג יד, בחדר הבחינה או בסמוך לו, כל חומר הקשור לבחינה/לקורס פרט לחומר שהשימוש בו הותר בכתב על ידי המורה.
4. יש למלא את הפרטים על מחברת הבחינה במקום המיועד לכך בלבד. אין לכתוב את השם או כל פרט מזהה אחר בתוך המחברת.
5. יש להישמע להוראות המשיג. נבחן לא יעזוב את מקומו ללא קבלת רשות המשיג. הפונה בשאלה או בבקשה ירים את ידו.

לשימוש המורה הבוחן:

| |
|---------------------------------|
| הציון <u>100</u> |
| המחברת נבדקה ביום <u>8/7/03</u> |
| חתימת המורה <u>3</u> |

סמסטר ב' תשס"ג
מועד: א' 22/6/2003
משך הבחינה: 3 שעות
חומר עזר: שני דפי עזר

בחינה בקורס: תוכנה 1
מרצה: פרופ' דן הלפרין

הנחיות כלליות לבחינה:

- המבחן מורכב משלוש שאלות תכנות ושאלה אמריקאית אחת.
- חובה לתעד את שאלות התכנות.
- נא לכתוב בכתב קריא ובאופן מסודר.
- מספר סטודנט: _____

בהצלחה !

שאלה 1 (25 נקודות)

כתוב את הפונקציה `char *letters (const char *str)` הפונקציה מקבלת מחרוזת ומחזירה מחרוזת המורכבת מן התווים שבמחרוזת המקורית, על פי סדר הופעתם, כך שאף תו אינו חוזר פעמיים. (שים לב שאין לדרוך על מחרוזת הקלט `str`).

למשל, אם הפרמטר `str` הינו המחרוזת:

Hello everybody how are you?

הרי שהערך המוחזר יהיה המחרוזת:

Helo vrybdhwau?

שאלה 2 (30 נקודות)

כתוב פונקציה בשם `create_matrix()` שמקבלת תמונה המורכבת מפיקסלים, ואת מימדיה. הפונקציה מחזירה את התמונה בייצוג שונה, כמתואר להלן, או NULL במקרה של כשלון.

עליך להשתמש במבנים הבאים:

```
typedef unsigned char Pixel[3];
```

```
typedef struct element {  
    struct element * left;  
    struct element * top;  
    struct element * right;  
    struct element * bottom;  
    Pixel pixel;  
} Element;
```

להלן ה- `prototype` של הפונקציה:

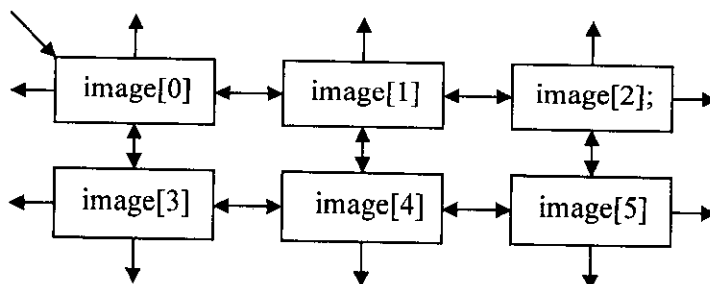
```
Element * create_matrix(Pixel * image, int width, int height);
```

הפונקציה מקבלת מצביע למערך חד מימדי בגודל `width * height` של אברים מסוג `Pixel`. השורות מופיעות בסדר עוקב בזיכרון. כל שורה היא בגודל `width`, וישנן `height` שורות.

הפונקציה מחזירה מצביע למבנה של רשימה מקושרת "דו-מימדית" המורכבת מרשומות מסוג `Element`. כל `Element` מכיל `Pixel`, ו-4 מצביעים ל-4 רשומות מסוג `Element` לכל היותר אשר מכילים את הפיקסלים משמאל, מלמעלה, מימין ומלמטה בהתאמה, אם קיימים, כפי שמתואר בציור. אם פיקסל לא קיים, המצביע המתאים הוא NULL.

לדוגמא, אם `width = 3` ו-`height = 2`, אזי הקלט ייראה כך:

| |
|----------|
| image[0] |
| image[1] |
| image[2] |
| image[3] |
| image[4] |
| image[5] |



והתוצאה תהיה:

שאלה 3 (35 נקודות)

נתון שלד של תכנית הקוראת מתוך קובץ בינארי מספר שלם n ו- $2n^2$ מספרי double שהם אברי שתי מטריצות ריבועיות A ו-B בגודל $n \times n$ כל אחת. התכנית מחשבת את מטריצת המכפלה C תוך שימוש בתהליך ילד נפרד עבור חישוב של כל איבר ב-C והעברת האינפורמציה לתהליך ההורה דרך pipe.

א. (15 נקודות)

כתוב הפונקציות `allocate_square_matrix()` (שמחזירה NULL בכשלון) ו-`get_input()` (שמחזירה 0 בסיום תקין ו-1 בכשלון).

אברי המטריצה A מופיעים בקובץ ברצף שורת מטריצה אחרי שורה ואחריהם באופן דומה מופיעים אברי B. אין צורך לכתוב את פונקציות העזר הנוספות. ניתן להניח שהן כתובות ועומדות לרשותנו. בפרט הפונקציה `multiply_ij()` המחשבת את האבר ה- $[i][j]$ במטריצת התוצאה.

ב. (20 נקודות)

השלם את ה-`main()`. ניתן להוסיף טיפוסים ומשתנים מעבר לאלו המופיעים בשלד.

```
/* use pipes to multiply two nxn matrices */
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

typedef double **matrix;

matrix allocate_square_matrix(int dimension);
int get_input(char *file_name, int *dimension, matrix *A, matrix *B);
double multiply_ij(matrix A, matrix B, int dimension, int i, int j);
void print_matrix(char *header, matrix A, int dimension);
void error_exit(char *s);

int main(int argc, char **argv)
{
    matrix A, B, C;
    int i, j;
    int n; /* square matrix dimension */
    int pd[2]; /* pipe descriptor */

    if (get_input(argv[1], &n, &A, &B) == -1)
        error_exit("get_input() failed");

    if (pipe(pd) < 0) /* create a pipe */
        error_exit("pipe() failed");

    /* complete this part of the main,
     * you may add more typedefs and variables
     */

    print_matrix("the matrix C", C, n);
    return 0;
}

matrix allocate_square_matrix(int dimension)
{
    /* complete */
}

int get_input(char *file_name, int *dimension, matrix *A, matrix *B)
{
    /* complete */
}
```

הקטע הבא הוא ה-makefile של תרגיל 6:

```

SPECFILE ?=spec.t
1 gen: gen.o
2 asm: asm.o inst.o
  sim: sim.o
  inst.o: inst.h
3 asm.o: inst.h
  sim.o: inst.h
4 inst.h inst.c: $(SPECFILE) gen
  gen $<
1 %.e : %.s asm
  asm $<
clean:
  rm -rf gen.o inst.o asm.o sim.o gen asm sim inst.c inst.h *.e

```

כשכותבים "ls -clt" (פקודה המדפיסה את תוכן ה-directory בסדר יורד של זמן עידכון אחרון), מקבלים:

```

-rw-r--r-- 1 efif math 167 May 23 22:47 spec.t
-rwxr-xr-x 1 efif math 21k May 23 22:45 sim*
-rw-r--r-- 1 efif math 9.6k May 23 22:45 sim.o
-rw-r--r-- 1 efif math 44 May 23 22:45 prog.e
-rwxr-xr-x 1 efif math 23k May 23 22:45 asm*
-rw-r--r-- 1 efif math 11k May 23 22:45 asm.o
-rw-r--r-- 1 efif math 1.8k May 23 22:45 inst.o
-rw-r--r-- 1 efif math 179 May 23 22:45 inst.c
-rw-r--r-- 1 efif math 618 May 23 22:45 inst.h
-rwxr-xr-x 1 efif math 25k May 23 22:34 gen*
-rw-r--r-- 1 efif math 14k May 23 22:34 gen.o
-rw----- 1 efif math 2.3k May 23 22:24 asm.c
-rw----- 1 efif math 3.6k May 23 22:24 gen.c
-rw----- 1 efif math 247 May 23 22:24 makefile
-rw-r--r-- 1 efif math 1.2k May 23 22:24 sim.c
-rw-r--r-- 1 efif math 30 May 10 10:54 prog.s

```

מה נקבל כשנכתוב "make prog.e sim"? בחר תשובה אחת מהבאות ותן הסבר קצר לבחירה (במחברת הבחינה).

| | |
|---|--|
| <p>1.</p> <pre> gcc -g -c -o gen.o gen.c gcc -g gen.o -o gen gen spec.t gcc -g -c -o asm.o asm.c gcc -g -c -o inst.o inst.c gcc -g asm.o inst.o -o asm asm prog.s gcc -g -c -o sim.o sim.c gcc -g sim.o -o sim </pre> | <p>3.</p> <pre> gcc -g -c -o asm.o asm.c gcc -g -c -o inst.o inst.c gcc -g asm.o inst.o -o asm asm prog.s gcc -g -c -o sim.o sim.c gcc -g sim.o -o sim </pre> |
| <p>2.</p> <pre> gen spec.t gcc -g -c -o asm.o asm.c gcc -g -c -o inst.o inst.c gcc -g asm.o inst.o -o asm asm prog.s gen spec.t gcc -g -c -o sim.o sim.c gcc -g sim.o -o sim </pre> | <p>4.</p> <pre> gen spec.t gcc -g -c -o asm.o asm.c gcc -g -c -o inst.o inst.c gcc -g asm.o inst.o -o asm asm prog.s gcc -g -c -o sim.o sim.c gcc -g sim.o -o sim </pre> |

char *letters (const char *str)

25
25

1 x 25

```
int i; /* loop counter */
char asc[256]; /* indication which characters we already got */
int len; /* the input string length */
char *ret; /* the return string */
int ret_len; /* the length of the return value */
```

```
len = strlen(str);
```

```
ret = (char *) malloc ((len+1) * sizeof(char)); /* allocating memory and checking pointer */
if (ret == NULL)
    return NULL;
```

```
ret_len = 0;
```

```
for (i=0; i<256; i++) {
    asc[i] = 0;
}
```

} /* initializing the "asc" array to be all zero's */

```
for (i=0; i<len; i++) {
    if (asc[str[i]] == 0) {
        ret[ret_len] = str[i];
        asc[str[i]] = 1;
        ret_len++;
    }
}
```

} /* main loop: running over the input string, if we encounter a "new" character we add it to the output string and change its value in the asc array to be 1 */

```
ret[ret_len] = '\0'; /* adding the \0 at the end of the output string */
return ret;
```

}

הפונקציה מקבלת שתי פרמטרים: const char *str ו- char *ret. היא מחזירה את האותיות היחידות מהstring str.

1. בחינה: האם האות str[i] כבר נמצאת ב-ret? אם לא, נוספת אותה ל-ret ונמנעת אותה ב-asc. אם כן, מתעלמים ממנה.
2. בחינה: האם הגענו לסוף string str? אם כן, מוסיפים את ה-\0 ל-ret ומחזירים את ret.
3. בחינה: האם ret היא NULL? אם כן, מחזירים NULL.

Element *create-matrix (Pixel *image, int width, int height) {

```

int i; /* loop counter */
Element *ret; /* the return pointer */
int len; /* the length of the image a/
assert (width > 0);
assert (height > 0);
len = width * height; /* setting the length of the array */

ret = (Element *) malloc (len * sizeof (Element)); /* memory allocation
if (ret == NULL) { /* for the new
return NULL; /* picture format */

```

```

for (i = 0; i < len; i++) { /* running over all the pixels */

    ret[i].pixel[0] = image[i][0];
    ret[i].pixel[1] = image[i][1];
    ret[i].pixel[2] = image[i][2]; /* setting the pixels
    /* RGB values */

    if ((i % width) == 0) {
        ret[i].left = NULL;
    } else {
        ret[i].left = &(ret[i-1]); /* if left pixel exists
    /* set the left pointer
    /* to point it, if
    /* not set it do
    /* be null */

    if ((i % width) == (width - 1)) {
        ret[i].right = NULL;
    } else {
        ret[i].right = &(ret[i+1]); /* same for right
    /* pointer

    if ((i / width) == 0) {
        ret[i].top = NULL;
    } else {
        ret[i].top = &(ret[i-width]); /* same for top
    /* pointer

    if ((i / width) == (height - 1)) {
        ret[i].bottom = NULL;
    } else {
        ret[i].bottom = &(ret[i+width]); /* same for
    /* bottom
    /* pointer
    }
}

```

return ret;

1. הפונקציה מקבלת את ה- image, width, height ו- len * sizeof (Element) כפרמטרים.
 2. הפונקציה מחזירה NULL אם יש שגיאה.
 3. הפונקציה מחזירה את המערך (array) של הפיקסלים (pixels) אם הכל תקין.

המחלקה הכלכלית

- * allocating main pointers
- pointer */

allocating
a doubles
array in
each cell

מִיִּצְלֹחַ וְעַד-כֹּכָב

$$\frac{8}{8}$$

מכון תל אביב-יפו

} $\frac{1}{\sqrt{2}}$ $\frac{1}{\sqrt{2}}$
 } $\frac{1}{\sqrt{2}}$ $\frac{1}{\sqrt{2}}$

dimension

$\frac{1}{2} \times \frac{1}{2} = \frac{1}{4}$
 $\frac{1}{4} \times \frac{1}{4} = \frac{1}{16}$
 $\frac{1}{16} \times \frac{1}{16} = \frac{1}{256}$
 $\frac{1}{256} \times \frac{1}{256} = \frac{1}{65536}$
 $\frac{1}{65536} \times \frac{1}{65536} = \frac{1}{4294967296}$

[illegible]

/★ closing the file ★/

return 0;

Pro 3

int pid;
int x, y;
double tmp;

fork
write
read

fork cc vpa op

for (i=0; i<n; i++) {

for (j=0; j<n; j++) {

pid = fork();

if (pid == 0)

{
if (write(p[1], &i, sizeof(int)) != sizeof(int))
error_exit("error in pipe");

if (write(p[1], &j, sizeof(int)) != sizeof(int))
error_exit("error in pipe");
tmp = multiply_2j(A, B, n, i, j);
if (write(p[1], &tmp,
sizeof(double)) != sizeof(double))
error_exit("error in pipe");

return 0;

}

for (i=0; i<n; i++) {

for (j=0; j<n; j++) {

if (read(p[0], &x, sizeof(int)) != sizeof(int))
error_exit("error in pipe");

if (read(p[0], &y, sizeof(int)) != sizeof(int))
error_exit("error in pipe");

if (read(p[0], &(C[x][y]), sizeof(double)) != sizeof(double))
error_exit("error in pipe");

close(p[0]);
close(p[1]);

ppe - a file j, i - new process

התהליך יקרא את הנתונים מהקובץ ויחשב את התוצאה

write
read
fork
error_exit

write
read
fork
error_exit

התהליך יקרא את הנתונים מהקובץ ויחשב את התוצאה

התהליך יקרא את הנתונים מהקובץ ויחשב את התוצאה

התהליך יקרא את הנתונים מהקובץ ויחשב את התוצאה

מחלקת 4
 $\frac{10}{10}$
 תשובות הנכונות הן 4.

4 בסדר:

בבקשה gen spect לבסס אתה ולכן תשובה הנכונה היא

3.

מכיוון שהבקשה gen spect היא גורמת בעצמה היא לא
 make, תשובה הנכונה היא 'א' או 'ב'.

gen. o. זה ייתכן gen. c. ולכן אין צורך לבקשה זאת ולכן
 תשובה 1 היא נכונה.