

הוראות לנבחנים ולנבחנות (נכתבו בלשון זכר אך נועדו לשני המינים)  
לפני התחלת הבחינה סלא את כל הפרטים הבאים בכתב ברור וקרא בעיון את ההוראות:

## הפקולטה למדעים מדויקים ע"ש ריימונד וברלי סאקלר

- הנך נדרש לשמור על טוהר הבחינה ועל עבודה עצמית ולהישמע להוראות המשגיחים ולנוהלי האוניברסיטה. אין להעתיק, אין לדבר ואין להעביר חומר בין הנבחנים.

נבחן הנוהג בניגוד להוראות צפוי להפסקת בחינתו ולהעמדה לדין משמעת.

- על הנבחן להבחן בחדר שבו הוא רשום.
- אין להחזיק טלפונים ניידים או אמצעי תקשורת ומכשירים אלקטרוניים כלשהם בזמן הבחינה. על הנבחן להניח את כל חפציו האישיים בצד החדר הרחק ממקום מושבו.

- אין להחזיק בהישג יד, בחדר הבחינה או בסמוך לו, כל חומר הקשור לבחינה או לקורס פרט לחומר שהשימוש בו הותר בכתב על ידי המורה.

- קריאת השאלון מותרת רק לאחר קבלת רשות מהמשגיח.
- נבחן לא יעזוב את מקומו ולא את חדר הבחינה בטרם סיים את הבחינה ללא קבלת רשות מהמשגיח. בעת יציאה מן החדר, יפקיד הנבחן את מחברות הבחינה והשאלון (טופס הבחינה) בידי המשגיח.

- נבחן שנכנס לחדר הבחינה וקיבל את השאלון לידי, לא יחזיר רשאי לעזוב אותו אלא כעבור חצי שעה לפחות מסועד תחילתה ורק לאחר שיחזיר למשגיח את המחברת ואת השאלון, ויקבל ממנו את התעודה המזהה שאותה מסר עם כניסתו לכיתה. נבחן שהחליט לעזוב בלי לכתוב את הבחינה יחשב כמי שנבחן במועד זה וציונו יהיה "0".

- אין לכתוב את השם או כל פרט מזהה אחר בתוך המחברת. פרטי הנבחן ימולאו על כריכת המחברת במקום המיועד לכך בלבד.

- אין לתלוש דפים מהמחברת. טיוטה תיכתב בתוך המחברת בלבד. אין להשתמש בדפים שהביא הנבחן.

- יש לכתוב את התשובות בעט כחול או שחור, בכתב יד ברור ונקי. בתום הבחינה יחזיר הנבחן את המחברת והשאלון ויקבל מיד המשגיח את התעודה המזהה.

11. אין לכתוב מעבר לקו האדום משני צידי הדף.

בהצלחה.

תאריך הבחינה 7.7.14

שם הקורס פיראט אונג

שם המורה פיראט רזא שן

מס' זיהוי  
(העתק מכרטיס הנבחן/התלמיד)

2	0	1	4	4	3	4	2	1
---	---	---	---	---	---	---	---	---



לשימוש המנהל והנבחן:

הציון \_\_\_\_\_

המחברת נבדקה ביום \_\_\_\_\_

חתימת המורה \_\_\_\_\_

4690065



50  
50  

---

100

ת.ז. 2014 3421  
מס. מחברת 01

משך הבחינה: שעתיים  
חומר עזר: שני דפי עזר בכתב יד או דף אחד מודפס

בחינה בקורס: פרויקט תוכנה, סמסטר ב' תשע"ד  
מרצים: פרופ' רודד שרן וגב' שיר פייביש-לנדאו

הנחיות כלליות לבחינה:

- בראש העמוד הנ"ל יש לציין את מספר תעודת הזהות ומספר המחברת.
- בבחינה חמש שאלות (פתוחות ואמריקאיות) בעלות ניקוד משתנה, בסך של 100 נק'.
- את התשובות לשאלות 2-5 יש למלא בטבלה המיועדת לכך (בעמוד זה). השאלות מניחות את קיום כל ההגדרות הדרושות להידור (קומפילצית) קטעי הקוד.
- חובה **לתעד** את התשובה לשאלה 1 (כמובן מותר בעברית). בתשובה יש לכלול את כל ההכרזות הדרושות, אך אין צורך להוסיף לקטעי הקוד פקודות #include.

בהצלחה !

טבלת תשובות לשאלות 2-5

שאלה 2	✓
שאלה 3	✓
שאלה 4	✓
שאלה 5	✓

© כל הזכויות שמורות  
מבלי לפגוע באמור לעיל, אין להעתיק, לצלם, להקליט, לשדר, לאחסן במאגר מידע, בכל דרך שהיא, בין מכונית ובין אלקטרונית או בכל דרך אחרת כל חלק שהוא מטופס הבחינה.





שאלה מס' 1 (50 נק')

כתוב תכנית המקבלת כארגומנטים ב-command line שם קובץ קלט ושם קובץ פלט. קובץ הקלט מכיל מספר כלשהוא של ספרות 0 ו-1 המיוצגות כתווים. על קובץ הפלט להכיל את אותו מידע (אפסים ואחדות) דחוס (בייצוג בינארי), כך שכל בית יכיל למעשה 8 ספרות לפי הדוגמא למטה. יש לבדוק שהקלט חוקי אך ניתן להניח שהוא מכיל מספר בתים המתחלק ב-8. לשם יעילות הקוד יש לכתוב את כל הפלט בבת אחת לקובץ (ולא למשל בית בודד בכל פעם).

דוגמא: אם קובץ הקלט מכיל 0 0 0 1 0 0 0 0 (מימין לשמאל) אז קובץ הפלט יכיל בית בודד שערכו הוא 16 (בייצוג בינארי).

שאלה מס' 2 (15 נק')

נתון קטע הקוד הבא:

```
int fun1(int* a, int* b)
{
    int i = 0;
    i += (&a == &b ? 1:0);
    i += (a == b ? 1:0);
    i += (*a == *b ? 1:0);
    return i;
}

int fun2(int** a, int* b)
{
    int i = 0;
    i += (a == &b ? 1:0);
    i += (*a == b ? 1:0);
    return i;
}

int main(void)
{
    int i = 0;
    int *p = &i;
    int *q = &i;
    printf("%d\n", fun1(p, q) + fun2(&p, q));
    return 0;
}
```

מהו פלט התוכנית?

1. א.
2. ~~ב.~~
3. ~~ג.~~
4. ד.
5. ה.

© כל הזכויות שמורות

מבלי לפגוע באמור לעיל, אין להעתיק, לצלם, להקליט, לשדר, לאחסן במאגר מידע, בכל דרך שהיא, בין מכונית ובין אלקטרונית או בכל דרך אחרת כל חלק שהוא מטופס הבחינה.





201443421

מס. ת.ז.

91

מס. מחברת

שאלה מס' 3 (10 נק')

נתון קטע הקוד הבא:

```

void reduceP(char* p, int i)
{
    while (i--){
        p--;
    }
}

int main(void)
{
    char* strings[] = {"abcdb", "bbb", "dddd"};
    char* p = *strings;
    char res[2];
    p += *strings[2] - *strings[1];
    res[0] = *p;
    reduceP(p, 2);
    p += 3;
    res[1] = *p;
    printf("%s\n", res);
    return 0;
}

```

מהו פלט התוכנית?

- א. cb
- ב. bb
- ג. c
- ד. b

© כל הזכויות שמורות

מבלי לפגוע באמור לעיל, אין להעתיק, לצלם, להקליט, לשדר, לאחסן במאגר מידע, בכל דרך שהיא, בין מכנית ובין אלקטרונית או בכל דרך אחרת כל חלק שהוא מטופס הבחינה.





שאלה מס' 4 (10 נק')

איזה בעיות קיימות או עלולות להיווצר בקוד הבא?

```

void foo2(char* str_a, char* str_b)
{
    int i = 0;
    while(str_a[i] != '\0')
        str_b[i] = str_a[i++];
    str_b[i] = '\0';
    return;
}

void foo1(char *str2)
{
    char *str1;
    str1 = (char*)malloc(20 * sizeof(char));
    if (str1 == NULL) return;
    if (str2 == NULL) return;
    if (scanf("%s", str1) != 1)
    {
        free(str1);
        return;
    }
    foo2(str1, str2);
    free(str1);
    return;
}

```

- (א) Memory corruption, buffer overflow attack  
 (ב) Memory leak, buffer overflow attack  
 (ג) Memory corruption, format string attack  
 (ד) Memory corruption, memory leak, buffer overflow attack  
 (ה) Memory corruption, memory leak, buffer overflow attack, format string attack

© כל הזכויות שמורות

מבלי לפגוע באמור לעיל, אין להעתיק, לצלם, להקליט, לשדר, לאחסן במאגר מידע, בכל דרך שהיא, בין מכנית ובין אלקטרונית או בכל דרך אחרת כל חלק שהוא מטופס הבחינה.





שאלה מס' 5 (15 נק')

נתונה רשימה מקושרת של תווים ופונקציה לבניית מתוך מחרוזת שאינה ריקה (חלקים מהקוד מוסתרים בכוכביות). איזו מהשורות א-ד תבטיח את בנייתה של רשימה מקושרת תקינה המכילה את כל תווי המחרוזת בסדר הפוך – כלומר התו האחרון במחרוזת הוא הראשון ברשימה וכן הלאה (בהנחה שכל הקצאות הזיכרון מצליחות):

```
struct list {
    char data;
    struct list *next;
};

struct list *build_reverse_list( char *str ) {
    struct list *head = (*****)malloc(*****);
    struct list *tail = head;
    tail->next = NULL;
    while( 1 ) {
        head->data = *str;
        if( !str[1] )
            break;

        /*א*/ tail->next=(struct list*)malloc(sizeof(struct list));tail=tail->next;
        /*ב*/ tail->next=(struct list*)malloc(sizeof(struct list *));tail=tail->next;
        /*ג*/ head=(struct list*)malloc(sizeof(struct list));head->next=tail;
        /*ד*/ head=(struct list*)malloc(sizeof(struct list));head->next=tail; tail=head;

        str++;
    }
    return head;
}
```

בהצלחה!

© כל הזכויות שמורות

מבלי לפגוע באמור לעיל, אין להעתיק, לצלם, להקליט, לשדר, לאחסן במאגר מידע, בכל דרך שהיא, בין מכונית ובין אלקטרונית או בכל דרך אחרת כל חלק שהוא מטופס הבחינה.





```
#include <stdlib.h>
#include <stdio.h>
char compress_8_chars(char *arr);
int check_file_is_valid(FILE *ifp);
void compress_array(char *dest, char *src, char *h);
```

OK

~~/\* gets a file and return the number of chars in it \*/~~

```
unsigned int get_char_num(FILE *ifp) {
    int i = fseek(ifp, 0, SEEK_END);
    return (int) (assert(i == 0);
    if (i != 0) return return (unsigned int) (ftell(ifp) - 1);
}
```

~~/\* compress 8 chars from \*arr to \*(arr+7) to one char \*/~~  
~~Precondition: \*arr is valid~~  
~~char compress\_8\_chars(char \*arr) {~~

```
    char c = 0; int i = 0;
    for (i = 0; i < 8; i++)
        c = (*arr++ == '0') ? 0 : 1;
    for (i = 1; i < 8; i++)
        c <<= 1;
    c |= ((*arr + i) == '0') ? 0 : 1;
    return c;
}
```

~~/\* gets file pointer to the beginning of file and checks that it's valid - if so returns char number otherwise EOF \*/~~

```
int check_file_is_valid(FILE *ifp) {
    int i = fseek(ifp, 0, SEEK_SET);
    char ch = 0;
    for (count = 0; (ch != EOF) && ((ch == '0') || (ch == '1')); ++count, ch = fgetc(ifp));
    assert(ferror(ifp) == 0);
    if (ch == EOF) return count - 1; // without EOF - not an error but EOF because of the assert
    return EOF;
}
```

~~/\* compress src to dest \*/~~

```
void compress_array(char *dest, char *src, char *h) {
    int i = 0, n = h / 8;
    for (i = 0; i < n; i++)
        *dest++ = compress_8_chars((src + i * 8));
}
```

```
int main(int argc, char *argv[]) {
    FILE *ifp, *ofp; int char_num = 0; char *arr, *compressed; int i = 0;
    assert(argc == 3);
    ifp = fopen(argv[1], "rb"); assert(ifp);
    ofp = fopen(argv[2], "wb"); assert(ofp);
    char_num = check_file_is_valid(ifp); assert(char_num != EOF);
    arr = (char *) malloc(char_num, sizeof(char)); assert(arr);
    compressed = (char *) malloc(char_num / 8, sizeof(char)); assert(compressed);
    fread(arr, sizeof(char), char_num, ifp); assert(ferror(ifp) == 0);
    compress_array(compressed, arr, char_num);
    fwrite(compressed, sizeof(char), char_num / 8, ofp); assert(ferror(ofp) == 0);
    free(arr); free(compressed); fclose(ifp); fclose(ofp); return 0;
}
```

1 > file

10 2

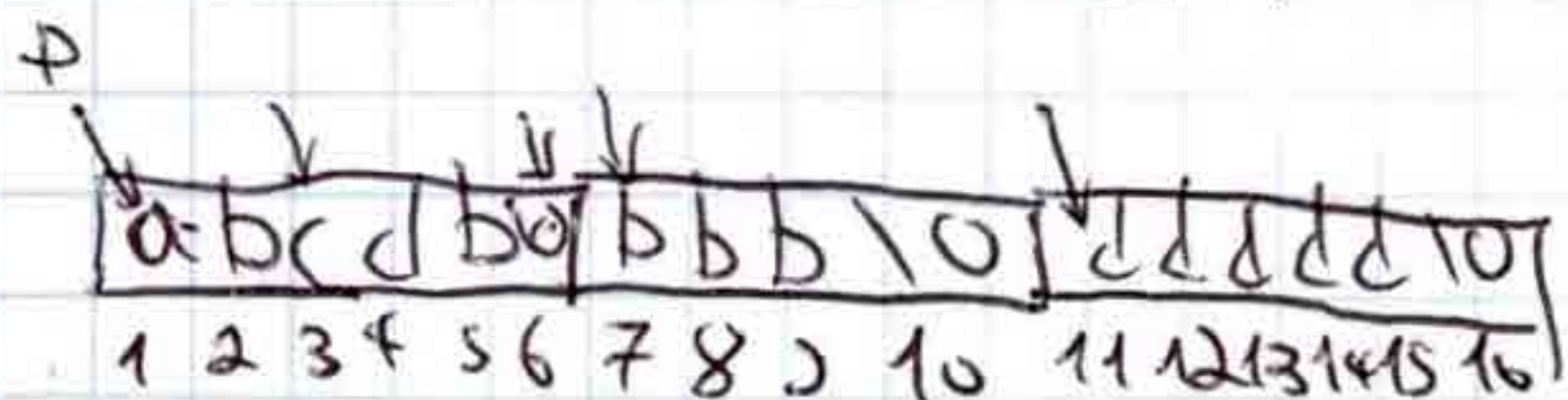
מכון הנדסת תוכנה



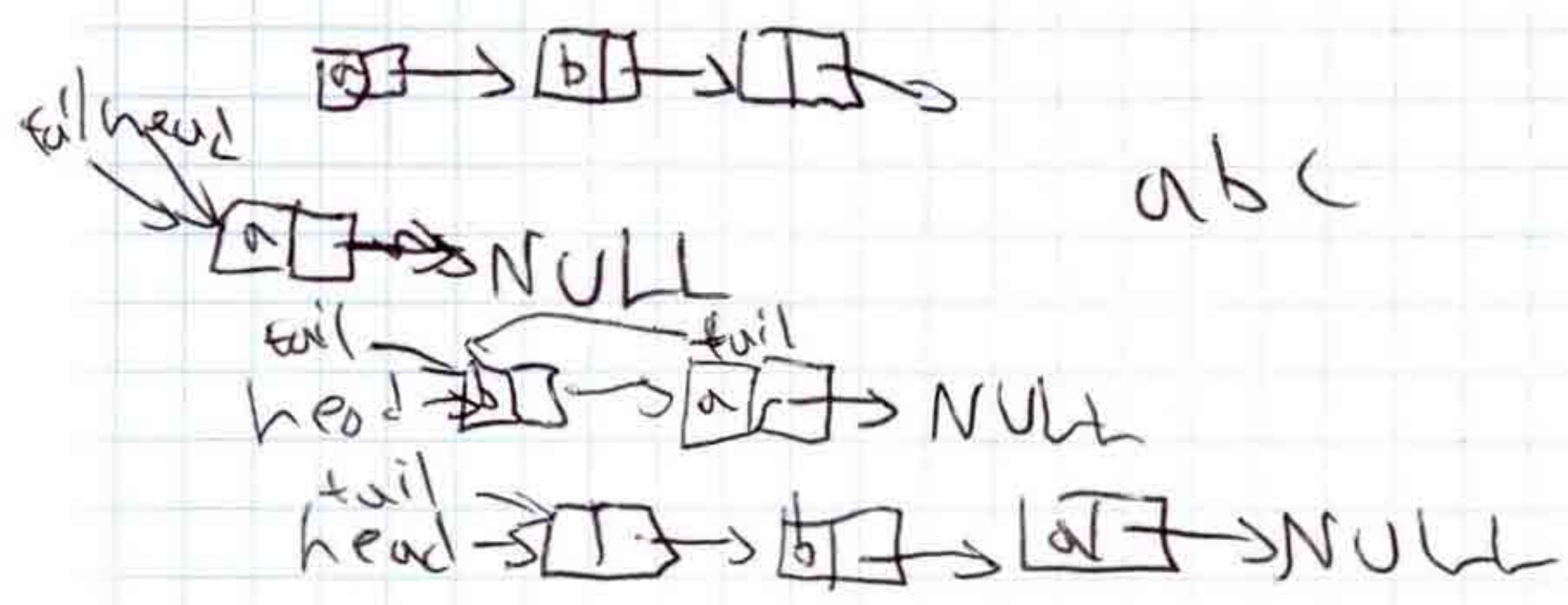




$i=0$   
 $p, a = \&i$   
 Fun1  $i=1$  2 return 2  
 $11-7=4$

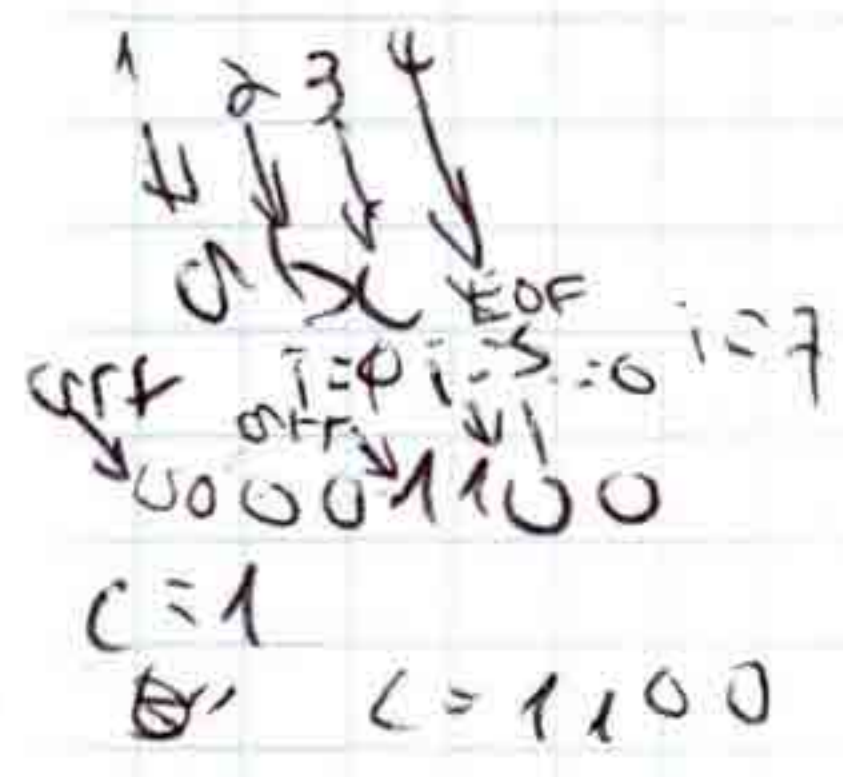


$P+4$



abc

$b = \&a$   
 $p = \&a$   
 $a = \&a$   
 $i=0$









// declarations

```
char compress_8_chars(char *arr);  
int get_char_h(FILE *fp);  
void compress_array(char *dest, char *src, int char_h);
```

1 > Re

/\* compress 8 chars from \*arr to \*(arr+7) into 1 char and returns it \*/

```
char compress_8_chars(char *arr) {
```

```
    int i; char c;  
    c = ((*arr) == '0') ? 0 : 1;  
    for (i = 1; i < 8; ++i) {  
        c <<= 1;  
        c |= ((*arr + i) == '0') ? 0 : 1;  
    }
```

```
    return c;  
}
```

so/so

/\* gets a pointer to a file and checks if it's valid.  
if valid returns the char number, if not returns EOF \*/

```
int get_char_h(FILE *fp) {
```

```
    int count = 0; i;  
    char ch = '0';  
    i = fseek(fp, 0, SEEK_SET);  
    assert(i == 0);  
    for (; ch != EOF && (ch == '0' || ch == '1'); ++count, ch = fgetc(fp));  
    assert(ferror(fp) == 0);  
    if (ch == EOF) return count - 1; // len without EOA  
    return EOF;
```

סיב' 1 < 8  
→ 00000000  
00000001  
00000010  
00000100  
00001000  
00010000  
00100000  
01000000  
10000000  
11111111

/\* compress src to dest - both should be in right size and dest-allocated \*/

```
void compress_array(char *dest, char *src, int char_h) {
```

```
    int i, n = char_h / 8;  
    for (i = 0; i < n; ++i) {  
        *(dest + i) = compress_8_chars(src + i * 8);  
    }
```

```
int main(int argc, char *argv[]) {
```

```
    FILE *ifp, *ofp; int char_num; char *arr, *compressed;  
    assert(argc == 3);  
    ifp = fopen(argv[1], "r"); assert(ifp);  
    ofp = fopen(argv[2], "w"); assert(ofp);  
    char_num = get_char_h(ifp); assert(char_num != EOF);  
    if (char_num == 0) { fclose(ifp); fclose(ofp); return 0; } // empty file  
    arr = (char *) malloc(char_num * sizeof(char)); assert(arr);  
    compressed = (char *) malloc((char_num / 8) * sizeof(char)); assert(compressed);  
    fread(arr, sizeof(char), char_num, ifp); assert(ferror(ifp) == 0);  
    compress_array(compressed, arr, char_num);  
    fwrite(compressed, sizeof(char), (char_num / 8), ofp); assert(ferror(ofp) == 0);  
    free(arr);  
    free(compressed);  
    fclose(ifp);  
    fclose(ofp);  
    return 0;
```

































לשימוש המרצה בלבד

טבלה לחישוב ציונים

[illegible]