



מכון טכנולוגי חולון  
Holon Institute of Technology

## המחלקה למדעי המחשב COMPUTER SCIENCE DEPARTMENT

סדנה מתקדמת בתכנות 61108  
סמסטר ב' תש"ף

### תרגיל מיוחד

### קבצים בינאריים, סיביות, מצביעים לפונקציות

שאלה 1 (30 נק'). נתונה ההגדרה הבאה לרשומה הכוללת בתוכה שדות לקוד אישי, שם ומשכורת העובד:

```
typedef struct employee
{
    int code;
    char name[15];
    float salary;
} Employee;
```

א. כתבו פונקציה שיוצרת **קובץ בינארי** של רשומות Employee. הפונקציה מקבלת כפרמטר את המחרוזת המהווה את שם הקובץ. על הפונקציה לארגן את קלט נתוני העובדים ולשמור אותם בקובץ. בסוף הקלט יש לקלוט את תו סיום הקובץ stdin.

ב. כתבו את הפונקציה הבאה:

```
void updateSalary(char* filename, float threshold).
```

הפונקציה מקבלת את שם קובץ העובדים הנ"ל (filename) ושכר הסף (threshold). ניתן להניח שכל המשכורות בקובץ נמוכות מהסף. על הפונקציה לארגן את קלט המספרים הממשיים המהווים תוספות, בהתאמה, למשכורותיהם של כל העובדים הרשומים בקובץ. המשכורות בקובץ יעלו בהתאם לתוספות הנקלטות. בנוסף, הרשומות בהן המשכורת תעלה מעל לשכר הסף **יימחקו מהקובץ (לא יוחלפו במקומות ריקים או באפסים אלא יימחקו פיסית)**. סדר הרשומות הנותרות יוותר כפי שהיה.

דוגמא. התוכן המקורי של הקובץ הוא

```
1214 "Moshe" 4500
9045 "Nadav" 4800
3011 "Ezra" 4100
5980 "Shmuel" 4600
```

ו- threshold=5000.

התוספות הנקלטות: 350, 300, 400, 600 (משמאל לימין).

התוכן המעודכן של הקובץ יהיה

1214 "Moshe" 4900  
3011 "Ezra" 4700  
5980 "Shmuel" 4950.

ג. כתבו פונקציה המקבלת את שם קובץ העובדים הנ"ל ומציגה את תוכן הקובץ על המסך.

כתבו פונקציה void Ex1() שקוראת לפונקציה 1.א ליצירת הקובץ, קוראת לפונקציה 1.ב לעדכון, ומציגה את הקובץ המקורי ואת הקובץ המעודכן באמצעות הפונקציה 1.ג. אין צורך לקלוט את שם הקובץ. אפשר להגדיר אותו ישר בתוך הפונקציה.

**שאלה 2 (35 נק').** נתונה מחרוזת ספרות dig\_str אשר מממשת מספר שלם לא שלילי. התו הראשון במחרוזת יכול להיות '0' רק אם היא מורכבת מתו יחיד.

ניתן להקטין את כמות הבתים הדרושים לאחסון המחרוזת באמצעות קידוד כל תו ב- 4 סיביות באופן הבא:

0000	'0'
0001	'1'
0010	'2'
0011	'3'
0100	'4'
0101	'5'
0110	'6'
0111	'7'
1000	'8'
1001	'9'

א. כתבו את הפונקציה הבאה:

char\* codingToShortString(char\* dig\_str).

על הפונקציה ליצור ולהחזיר מחרוזת חדשה short\_dig\_str. כל בית ב- short\_dig\_str יורכב משתי רביעיות הסיביות המתאימות לשני תווים עוקבים של dig\_str. עבור dig\_str באורך n, אורכה של short\_dig\_str יהיה n/2. עבור n זוגי ו- n/2+1 עבור n אי-זוגי. עבור n אי-זוגי, הרביעייה הראשונה ב- short\_dig\_str לא מתאימה לאף תו של dig\_str וכל הסיביות בה יהיו אפסים.

**דוגמא:**

עבור dig\_str="1234", המחרוזת short\_dig\_str תורכב מסיביות הבאות:  
00010010 00110100

עבור dig\_str="51234", המחרוזת short\_dig\_str תורכב מסיביות הבאות:  
00000101 00010010 00110100

(משמאל לימין, מהסיבית המשמעותית ביותר – MSB - לסיבית הכי פחות משמעותית - LSB).

יש להקצות את המקום הנדרש בזיכרון למחרוזת short\_dig\_str בצורה מדויקת. ניתן להניח שיש בזיכרון מספיק מקום להקצאה.

ב. כתבו את הפונקציה הבאה:

```
void displayShortString(char* short_dig_str).
```

על הפונקציה להציג short\_dig\_str על המסך בשתי צורות: עשרונית ובינארית. כאשר אורך של המחרוזת המקורית ( dig\_str ) התואמת הוא אי-זוגי, אין להציג את 4 האפסים הראשונים בצורה בינארית של short\_dig\_str. יש להכניס רווחים בין רביעיות של הייצוג הבינארי.

למשל, בהמשך לדוגמאות הקודמות, הייצוגים האפשריים הם

```
1234 = 0001 0010 0011 0100
51234 = 0101 0001 0010 0011 0100
```

כתבו פונקציה void Ex2() שקולטת את המחרוזת dig\_str, יוצרת את המחרוזת short\_dig\_str באמצעות פונקציה 2.א, מציגה אותה באמצעות פונקציה 2.ב ומשחררת את הזיכרון. למחרוזות dig\_str אפשר להשתמש במערך סטטי בעל גודל מספק. ניתן להניח שיש במערך מספיק מקום לקלט המחרוזת ושהקלט הוא תקין.

שאלה 3 (35 נק'). נתון מערך תווים דו-ממדי בגודל 8x8 המוגדר באופן הבא:

```
#define SIZE 8
typedef char board[SIZE][SIZE];
board bin_board;
```

א. כתבו את הפונקציה הבאה:

```
void fillBoard(board bin_board).
```

על הפונקציה למלא את המערך bin\_board בתווי '0' ו-'1' באופן אקראי באמצעות הפונקציה rand() ולהציג את המערך על המסך.

ב. כתבו את הפונקציה הבאה:

```
unsigned long long boardToBits(board bin_board).
```

על הפונקציה להצפין את המערך במספר מסוג unsigned long long בעל 64 סיביות, כך שתו '1' ייוצג באמצעות סיבית 1 ותו '0' ייוצג באמצעות סיבית 0 (מ- bin\_board[0][0] ל- bin\_board[7][7], מה- MSB אל ה- LSB, בהתאמה). הפונקציה תחזיר את המספר.

ג. כתבו את הפונקציה הבאה:

```
void displayBinaryBoard(unsigned long long coded_bin_board).
```

על הפונקציה להציג על המסך את המספר המתקבל כטבלת אפסים ואחדות (סיבית 0 כתו '0', סיבית 1 כתו '1'). כל שמיניית סיביות תוצג בשורה נפרדת, השמינייה המשמעותית ביותר תהיה העליונה, השמינייה הכי פחות משמעותית תהיה התחתונה. במילים אחרות, על הייצוג להיות זהה לייצוג המערך ב- 3.א.

כתבו פונקציה void Ex3() שממלאת ומציגה את המערך באמצעות פונקציה 3.א, מצפינה את המערך ע"י פונקציה 3.ב ומציגה את הקוד דרך פונקציה 3.ג.

יש לכתוב תכנית שמפעילה את הפונקציות  $Ex1()$ ,  $Ex2()$ ,  $Ex3()$  הנ"ל.  
התוכנית תעבוד כתפריט לבחירת אחת משלוש הפונקציות (בדומה למטלות 1 ו-2 שבאתר הקורס).

### הוראות

1. יש להשתמש בשמות משמעותיים ובהערות.
2. יש להקפיד לכתוב בצורה מבנית.
3. יש להקפיד על שימוש בממשק ידידותי ככל האפשר. אין צורך בבדיקת תקינות הקלט.
4. תכנית שלא תעבור קומפילציה לא תתקבל!