

# שפת C – תרגיל 1

היכרות עם השפה, משתנים, לולאות, תנאים, פונקציות,  
קלט/פלט

תאריך הגשה:

הגשה מאוחרת (בהפחתת 10 נקודות):

הנחיות חשובות:

א. בכל התרגילים יש לעמוד בהנחיות הגשת התרגילים וסגנון כתיבת הקוד. שני המסמכים נמצאים באתר הקורס – הניקוד יכלול גם עמידה בדרישות אלו.

ב. אין להגיש קבצים נוספים על אלו שתדרשו

ג. עליכם לקמפל עם הדגל Wextra- ולוודא שהתוכנית מתקמפלת ללא אזהרות, **תכנית שמתקמפלת עם אזהרות תגרור הורדה בציון התרגיל**. למשל, בכדי לקמפל קובץ מקור בשם ex1.c יש להריץ את הפקודה:

```
gcc -Wextra ex1.c -o ex1
```

ד. עליכם לוודא שהתרגילים שלכם תקינים ועומדים **בכל דרישות הקימפול והריצה במחשבי בית הספר** מבוססי מעבדי bit-64 (מחשבי האקווריום, לוי, השרת river). חובה להריץ את התרגיל במחשבי בית הספר לפני ההגשה. (ניתן לוודא שהמחשב עליו אתם עובדים הנו בתצורת bit-64 באמצעות הפקודה "uname -a" ווידוא כי הארכיטקטורה היא 64, למשל אם כתוב x86\_64).

ה. לאחר ההגשה, בדקו את הפלט המתקבל בקובץ ה-PDF שנוצר מה submission script בזמן ההגשה. באם ישנן שגיאות, תקנו אותן על מנת שלא לאבד נקודות.

ו. **בדיקת הקוד לפני ההגשה, גם על ידי קריאתו וגם על ידי כתיבת בדיקות אוטומטיות (tests) עברו היא אחריותכם**

חישבו על מקרי קצה לבדיקת הקוד, בדיקות אוטומטיות יהיו חלק מבדיקת הקוד לציון קבצי בדיקה לדוגמה ניתן למצוא בתיקיה: ~labc/public/ex1/ex1\_tests.tar וקובץ בדיקות באתר הקורס: ex1\_tests.tar

**חלק מקבצי הבדיקה לדוגמה שמסופקים על ידי צוות הקורס נבדקים על ידי הבדיקה**

האוטומטית בהגשה (presubmission script). שימוש בקבצי הבדיקה לדוגמה הוא באחריותכם. במהלך הבדיקה הקוד שלכם ייבדק מול קלטים נוספים לשם מתן הציון.

### מידע חשוב נוסף:

1. ניתן להתחבר באמצעות SSH למחשבי בית הספר (למשל לשם בדיקת הקוד לפני הגשה מהבית)

[http://wiki.cs.huji.ac.il/wiki/Connecting\\_from\\_outside](http://wiki.cs.huji.ac.il/wiki/Connecting_from_outside)

2. עליכם להכיר את ספריית הקלט-פלט של שפת C ובייחוד את השימוש בפונקציות printf ו scanf

<http://www.cplusplus.com/reference/clibrary/cstdio>

3. תזכורת: כדי להשתמש בספריות של שפת C יש להכיל את הספרייה באמצעות הוספת קובץ ה-header המתאים. לדוגמה, כדי להשתמש בספריית הקלט-פלט יש לכתוב בראש הקוד של התכנית

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

### משימת תכנות - ספירת מילים (50 נק')

בתרגיל זה עליך לכתוב תוכנה שתמנה את השורות, המילים והתווים בקלט.

עם סיום הקלט על התוכנית להדפיס את שלושת המספרים, מופרדים ברווחים.

קובץ להגשה: בתרגיל זה עליכם להגיש את הקובץ count.c המכיל את התכנה המבוקשת. הערות:

1. שורה מסתיימת בתו '\n', מילה מסתיימת ברווח או בסוף השורה.

2. הקלט נגמר בתו EOF (end of file), שמוגדר בספרייה stdio.h, אין לספור את התו EOF עצמו (בספירת התווים)

3. כדי להכניס EOF בטרמינל יש להכניס ctrl+D.

4. יש לספור גם שורות ריקות.

5. בלינוקס קיים כלי בשם wc שעושה את הפעולה המדוברת, אתם יכולים להיעזר בו כדי לבדוק את התוכנית שלכם. אבל שימו לב שהפלט שלו שונה טיפה מהפלט שנדרש מכם בתרגיל זה. בכל מקרה של אי התאמה עליכם להתאים לפתרון בית הספר.

## משימת תכנות - צופן הזזה (50 נק')

צופן החלפה, הוא צופן בו כל אות בשפה מוחלפת באות אחת אחרת, תוך שמירה על סדר האותיות המקורי. צופן הזזה הוא צופן החלפה בו כל אות מוחלפת באמצעות הזזה שלה בא'ב'.

למשל, בצופן הזזה עם היסט 7, האות a תוחלף באות h, כי זו האות שבאה 7 אותיות אחריה בא'ב'.

בתרגיל זה עליך לכתוב שתי פונקציות שכותרתן:

```
char decrypt(int shift, char c);  
char encrypt(int shift, char c);
```

הפונקציה encrypt מקבלת תו והיסט, ומחזירה את התו המוצפן, כאשר:

א. אותיות אנגליות קטנות או גדולות: מוזזות בהיסט הנתון, מעגלית (אחרי z בא a עוד פעם).

ב. מספרים מוזזים ביניהם, מעגלית גם כן.

ג. תווים אחרים נשארים ללא שינוי.

הפונקציה decrypt עושה בדיוק את ההפך מהפונקציה encrypt – מפענחת טקסט מוצפן.

דוגמאות:

`encrypt(2, 'a') = 'c'`

`encrypt(1, 'Z') = 'A'`

`encrypt(3, '4') = '7'`

`encrypt(3, '%') = '%'`

`decrypt(12, '2') = '0'`

`'decrypt(13, 'F') = 'R'`

`decrypt(n, encrypt(n, c)) = c` (for every integer n and char c)

בנוסף עליכם לכתוב תוכנית קטנה שתדגים שימוש בפונקציות שכתבתם, על התוכנית לשאול את המשתמש מה ההיסט (להדפיס: "please enter the shift amount:") בשלב זה המשתמש יכניס את ההיסט (מספר בין 0 ל 25), לאחר מכן התוכנית תשאל את המשתמש אם ברצונו להצפין טקסט או לפענח טקסט מוצפן (להדפיס "would you like to encrypt (e) or decrypt (d)?"), בשלב

זה המשתמש יכניס e אם ברצונו להצפין ו d אם ברצונו לפענח. (אם המשתמש הכניס תו אחר על התוכנית לחזור ולבקש) אחרי זה המשתמש יכתוב את המחרוזת שצריך להצפין או לפענח (שורה אחת), אחרי שהמשתמש הכניס את המחרוזת התוכנית תדפיס את התוצאה - המחרוזת אחרי ההצפנה או הפענוח.

#### הערות:

- א. אתם רשאים להניח שמחרוזת הקלט אינה מכילה יותר מ 100 תווים.
- ב. שלוש הפונקציות (encrypt, decrypt, main) צריכות להיות מוגשות בקובץ shift.c

#### **diff**

diff הנה תוכנה להשוואה טקסטואלית של שני טקסטים שונים. בהינתן שני קבצי טקסט להשוואה, 1.txt ו-2.txt:  
הפקודה הבאה תדפיס את השורות אשר אינן זהות בשני הקבצים:

```
diff 1.txt 2.txt
```

במידה והקבצים זהים לחלוטין, לא יודפס דבר.

בכדי להשוות פלט של תכנית myProgram לפלט נתון expected.out ניתן לעבוד בשתי דרכים:  
1. ניתוב פלט התכנית אל קובץ ולאחר מכן השוואת הקובץ שנוצר לקובץ הנתון:

```
myProgram > myOutput.out
```

```
diff myOutput.out expected.out
```

2. השוואת קובץ נתון עם פלט ריצת התכנית ללא כתיבה לקובץ באמצעות piping:

```
myProgram | diff expected.out -
```

קראו על אפשרויות נוספות של diff בעזרת הפקודה man diff.

#### **הגשה**

עליכם להגיש קובץ tar בשם ex1.tar המכיל רק את שני הקבצים הבאים:

```
shift.c, count.c
```

ניתן ליצור קובץ tar כדרוש על ידי הפקודה:

```
tar cvf ex1.tar shift.c count.c
```

לפני ההגשה, פתחו את הקובץ ex1.tar בתיקיה נפרדת וודאו שהקבצים מתקמפלים ללא שגיאות וללא אזהרות.

מומלץ גם להריץ בדיקות אוטומטיות וטסטרים שכתבתם על הקוד אותו אתם עומדים להגיש. בנוסף, אתם יכולים להריץ בעצמכם בדיקה אוטומטית עבור סגנון קידוד בעזרת הפקודה:

```
~labc/public/codingStyleCheck <file or directory>
```

כאשר <directory or file> מוחלף בשם הקובץ אותו אתם רוצים לבדוק או תיקייה שיבדקו כל הקבצים הנמצאים בה.

דאגו לבדוק לאחר ההגשה את קובץ הפלט (submission.pdf) וודאו שההגשה שלכם עוברת את ה-presubmission script ללא שגיאות או אזהרות.

בהצלחה!