

# שפת C – תרגיל 1

## היכרות עם השפה, preprocessor, compiler, משתנים, לולאות, תנאים, פונקציות, קלט/פלט

תאריך הגשה: יום שני 2.11.15 עד שעה 23:55

הגשה מאוחרת (בהפחתת 10 נקודות): יום שלישי 2.11.15 עד שעה 23:55

תאריך ההגשה של הבוחן: יום שני 2.11.15 עד שעה 23:55

### 1. הנחיות חשובות:

1. בכל התרגילים יש לעמוד בהנחיות הגשת התרגילים וסגנון כתיבת הקוד. שני המסמכים נמצאים באתר הקורס – הניקוד יכלול גם עמידה בדרישות אלו.
2. בכל התרגילים עליכם לכתוב קוד ברור. בכל מקרה בו הקוד שלכם אינו ברור מספיק עליכם להוסיף הערות הסבר בגוף הקוד. יש להקפיד על תיעוד (documentation) הקוד ובפרט תיעוד של כל פונקציה.
3. במידה ואתם משתמשים בעיצוב מיוחד או משהו לא שגרתי, עליכם להוסיף הערות בקוד המסבירות את העיצוב שלכם ומדוע בחרתם בו.
4. בכל התרגילים במידה ויש לכם הארכה ואתם משתמשים בה, חל איסור להגיש קובץ כלשהוא בלינק הרגיל (גם אם לינק ההגשה באיחור טרם נפתח). מי שיגיש קבצים בשני הלינקים מסתכן בהורדת ציון משמעותית.
5. אין להגיש קבצים נוספים על אלו שתדרשו.
6. עליכם לקמפל עם הדגלים `Wall -Wextra -Wvla -std=c99` ולוודא שהתוכנית מתקמפלת ללא אזהרות, תכנית שמתקמפלת עם אזהרות תגרור הורדה בציון התרגיל. למשל, בכדי ליצור תוכנית מקובץ מקור בשם `ex1.c` יש להריץ את הפקודה:

`gcc -Wextra -Wall -Wvla -std=c99 ex1.c -o ex1`

7. 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"` ויודא כי הארכיטקטורה היא 64, למשל אם כתוב `x86_64`)
8. לאחר ההגשה, בדקו את הפלט המתקבל בקובץ ה-PDF שנוצר מהpresubmission script בזמן ההגשה. באם ישנן שגיאות, תקנו אותן על מנת שלא לאבד נקודות. שימו לב! תרגיל שלא יעבור את הpresubmission script ציונו ירד משמעותית (הציון יתחיל מ-50, ויוכל לרדת) ולא יהיה ניתן לערער על כך.
9. בדיקת הקוד לפני ההגשה, גם על ידי קריאתו וגם על ידי כתיבת בדיקות אוטומטיות (tests) עבורו היא אחראיתכם. חישבו על מקרי קצה לבדיקת הקוד. קבצי בדיקה לדוגמה ניתן למצוא פה: `~slabc/www/ex1/tests_examples.tar`
- שימוש בקבצים אלו הוא באחריותכם. במהלך הבדיקה הקוד שלכם ייבדק מול קלטים נוספים לשם מתן הציון.
10. הגשה מתוקנת - לאחר מועד הגשת התרגיל ירצו הבדיקות האוטומטיות ותקבלו פירוט על הטסטים בהם נפלתם. לשם שיפור הציון יהיה ניתן להגיש שוב את התרגיל לאחר תיקוני קוד קלים ולקבל בחזרה חלק

## 2. 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 scanf וprintf  
<http://www.cplusplus.com/reference/cstdio>

## 3. הנחיות ספציפיות לתרגיל זה:

1. בשאלה של Shift אין להשתמש בספריות כלשהן מלבד <stdio.h>.

## 4. ספירת מילים - Count (35 נק) :

1. בתרגיל זה עליכם לכתוב קובץ בשם Count.c המכיל תוכנה שתמנה את השורות, המלים והתווים בטקסט.  
עם סיום הקלט על התוכנית להדפיס את שלושת המספרים, באופן הבא.

“Num of rows:3 words:3 characters:9\n”

2. הערות:

- a. הקלט של התכנה הינו ה standard input
- b. הפלט של התכנה הינו ה standard output
- c. שורה מסתיימת 'ח' או בתו EOF
- d. מילה מסתיימת ברווח או בסוף השורה או בתו EOF
- e. הקלט מסתיים בתו EOF (end of file, שמוגדר בספרייה stdio.h, אין לספור את התו EOF עצמו)
- f. כדי להכניס EOF בטרמינל יש להכניס ctrl+D.
- g. יש לספור גם שורות ריקות.
- h. אינכם יכולים להניח הנחות כלשהן על אורך הקלט ותוכנו.
- i. על הפתרון שלכם להתנהג באופן תואם לפתרון בית הספר.

## 5. צופן הזזה - Shift (65 נק) :

1. צפני החלפה, המכונים גם צפנים חד-אלפביתיים, הם צפנים בהם מוחלפת כל אות בשפה באות אחרת. (תוך שמירה על סדר האותיות המקורי וכך שכל אות מקורית מיוצגת על ידי אות אחת ויחידה בטקסט המוצפן).
2. צופן הזזה הינו צופן החלפה בו כל אות מוחלפת באמצעות הזזה שלה בא'ב'.
3. לדוגמא: בצופן הזזה עם היסט 7, האות a תוחלף באות h (כי זו האות שבאה 7 אותיות אחריה בא'ב').  
עליכם לכתוב קובץ בשם Shift.c המכיל מימוש למתודות (decrypt, encrypt) המופיעות ב- header file שסופק לכם.
4. הפונקציה encrypt מקבלת תו והיסט, ומחזירה את התו המוצפן כאשר:  
a. אותיות באנגלית (גדולות או קטנות) מוזזות בהיסט הנתון בצורה מעגלית (אחרי z בא שוב a).  
b. מספרים מוזזים ביניהם מעגלית (אחרי 9 בא שוב 0)  
c. תווים אחרים נשארים ללא שינוי.
5. הפונקציה decrypt מבצעת את הפעולה ההפוכה, ומפענכת טקסט מוצפן באותו אופן.

דוגמאות:

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

```
encrypt(1, 'Z') => 'A'
encrypt(3, '4') => '7'
encrypt(3, '%') => '%'
decrypt(12, '2') => '0'
decrypt(13, 'F') => 'R'
```

שימו לב:

$\text{decrypt}(n, \text{encrypt}(n, c)) = c$  (for every integer  $n$  and char  $c$ )

6. בנוסף עליכם לכתוב בקובץ Shift.c תוכנית קטנה (שתקרא מהפונקציה main) שתדגים את השימוש בפונקציות שכתבתם.

a. על התוכנית לשאול את המשתמש מה ההיסט הרצוי:

"Please enter the shift amount:"

b. בשלב זה על המשתמש להכניס את ההיסט (מספר שלם בין 0 ל-50).

- אתם רשאים להניח שהקלט מהמשתמש יהיה מספר שלם (מסוג integer), אבל עליכם לוודא שהוא בטווח המבוקש. במידה ולא עליכם להדפיס הודעת שגיאה (ראו טסטים בדוגמאות) ולחזור לשלב הקודם.

c. לאחר מכן התוכנית תשאול את המשתמש האם ברצונו להצפין טקסט או לפענח טקסט מוצפן

"Would you like to encrypt (e) or decrypt (d)?"

d. בשלב זה המשתמש יכניס 'e' אם ברצונו להצפין ו-'d' אם ברצונו לפענח.

- אם המשתמש הכניס תו אחר, על התוכנית להדפיס הודעת שגיאה (ראו טסטים בדוגמאות) ולחזור שוב לשלב הקודם.

e. לאחר מכן המתמש יכתוב את המחרוזת שצריך להצפין או לפענח.

- שורה אחת. אתם רשאים להניח שמספר התווים שהמשתמש יקליד בשורה  $\Rightarrow$  100 תווים. במידה והמשתמש הקליד יותר מ-100 תווים בשורה, התעלמו משאר התווים.

f. עם סיום הקלט על התוכנית להדפיס את הקלט המקורי (בתוך גרשיים), רווח, חץ ימינה (">"), רווח, המחרוזת שפוענחה/הוצפנה (בתוך גרשיים) ולרדת שורה ("n").

דוגמא (צופן קיסר):

Please enter the shift amount:

3

Would you like to encrypt (e) or decrypt (d)?

e

hello

"hello" -> "khoor"

g. על פלט התוכנית שלכם לתאום במדויק את הפלטים של הטסטים לדוגמא שסיפקנו לכם.

h. אין להשתמש בספריות כלשהן מלבד `<stdio.h>`

## 7. הערות למשימות התכנות:

1. התכניות יבדקו גם על סגנון כתיבת הקוד וגם על פונקציונאליות, באמצעות קבצי קלט שונים (תרחישים שונים להרצת התכניות). הפלט של פתרונותיכם ישווה (השוואת טקסט) לפלט של פתרון בית הספר. לכן עליכם להקפיד על פורמט הדפסה מדויק, כדי למנוע שגיאות מיותרות והורדת נקודות.
2. לרשותכם כמה קבצי קלט לדוגמה וקבצי הפלט המתאימים להם (אלו מהווים רק חלק קטן מקבצי הקלט-פלט שנשתמש בהם, כתבו לעצמכם בדיקות נוספות). עליכם לוודא שהתכנית שלכם נותנת את אותו הפלט בדיוק.
3. על מנת לעשות זאת הריצו את תכניתכם עם הקלט לדוגמה על ידי ניתוב ה standard input להקרא מקובץ (באמצעות האופרטור "<" בשורת ההרצה ב terminal), ונתבו את הפלט של תכניתכם, שהוא ה standard output, לתוך קובץ (באמצעות האופרטור ">") באופן הבא:

```
ProgramName < inputFile > myOutputFile
```

4. השוו את קובץ הפלט שנוצר לכם עם קובץ הפלט המתאים של פתרון בית הספר, באמצעות הפקודה diff

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

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

במידה והקבצים זהים לחלוטין, לא יודפס דבר.

קראו על אפשרויות נוספות של diff בעזרת הפקודה man diff. לחלופין אתם יכולים גם להשתמש בתוכנה tkdiff אשר מראה גם את השינויים ויזואלית.

כמו כן, אתם יכולים גם להשוות ישירות באופן הבא:

```
ProgramName < inputFile | diff expected.out
```

5. אם ישנם מקרים שהוראות התרגיל לא מציירות בבירור כיצד התכנית צריכה להתנהג, הביטו בקבצי הקלט וקבצי הפלט לדוגמה שניתנים לכם ובדקו אם התשובה לשאלתכם נמצאת שם. כמו כן, היעזרו בפתרון בית הספר, הריצו עליו את הטסטים שלכם והשוו להתנהגות תוכניתכם.

## 8. קבצי עזר:

1. את הקבצים הנדרשים לצורך התרגיל ניתן למצוא ב:

```
~labc/www/ex1/ex1_files.tar
```

2. את פתרון הבית ספר ניתן למצוא ב:

```
~labc/www/ex1/schoolSol.tar
```

3. דוגמאות לטסטים ניתן למצוא ב:

```
~labc/www/ex1/tests_examples.tar
```

## 9. הגשה:

1. עליכם להגיש קובץ tar בשם ex1.tar המכיל רק את הקבצים הבאים:  
Shift.c, Count.c
2. ניתן ליצור קובץ tar כדרוש על ידי הפקודה:

```
tar cvf ex1.tar Shift.c Count.c
```

3. לפני ההגשה, פתחו את הקובץ ex1.tar בתיקה נפרדת וודאו שהקבצים מתקמפלים ללא שגיאות וללא אזהרות.
4. מומלץ מאוד 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/www/codingStyleCheck <file or directory>
```

כאשר <directory or file> מוחלף בשם הקובץ אותו אתם רוצים לבדוק או תיקייה שיבדקו כל הקבצים הנמצאים בה (שימו לב שבדיקה אוטומטית זו הינה **רק חלק** מבדיקות ה codingStyle)

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

```
~labc/www/ex1/presubmit_ex1 <tar file>
```

## בהצלחה!