האוניברסיטה העברית בירושלים

בית הספר להנדסה ולמדעי המחשב ע״ש רחל וסלים בנין

סדנת תכנות C ו- C++ - תרגיל 1

<u>נושאי התרגיל:</u> היכרות עם השפה, קומפילציה, משתנים, אריתמטיקה פשוטה, פלט, תנאים, לולאות, פונקציות ושימוש ב- CLI

תאריך הגשה: 5.11.2020

1 רקע

קריפטוגרפיה הוא תחום עתיק יומין, שלו ניתן למצוא תיעוד עוד לפני 500 שנים. בעבר, נעשה שימוש בקריפטוגרפיה בעיקר על ידי הצבא והמלוכה, בעוד היום זהו נושא שניתן אף לטעון שכל אחד מאיתנו עושה בו שימוש על בסיס יום יומי, ואף בכל שניה שאנו משתמשים במחשב האישי שלנו (או במכשיר החכם הנייד) – אף מבלי לשים לב לכך.

בתרגיל זה נממש תוכנה המצפינה ומקודדת טקסט באמצעות צופן הנקרא ״צופן קיסר״, או ״צופן היסט״.

2 צופן קיסר (צופן היסט)

נפתח בכך שנתאר כיצד פועל צופן קיסר באופן לא פורמלי: נסמן ב Σ את האלפבית ״שהמצפין״ יודע לקודד. המצפין מקבל מחרוזת כלשהי, s, וערך הזחה $k \in \mathbb{N} \cup \{0\}$, עבור כל תו $c \in s$ אם $c \in \Sigma$ המצפין יבצע ״הזחה ימינה״ של c, k פעמים. למשל, אם k=2 וקיבלנו את התו 'A' אזי נזיח אותו פעמיים – פעם ראשונה ל- 'B' ופעם שניה ל- 'C'. הערך 'C' הוא הערך שמתקבל, אפוא, מהצפנת התו 'A' עם k=2.

עתה, ננסה להיות קצת יותר פורמלים, ונגדיר את צופן קיסר באופן הבא:
זו הגדרה מצומצמת יותר מההגדרה המלאה של צופן קיסר, אך היא תשרת אותנו מהימנה בתרגיל זה.
למתעניינים, ראו
https://en.wikipedia.org/wiki/Caesar_cipher/

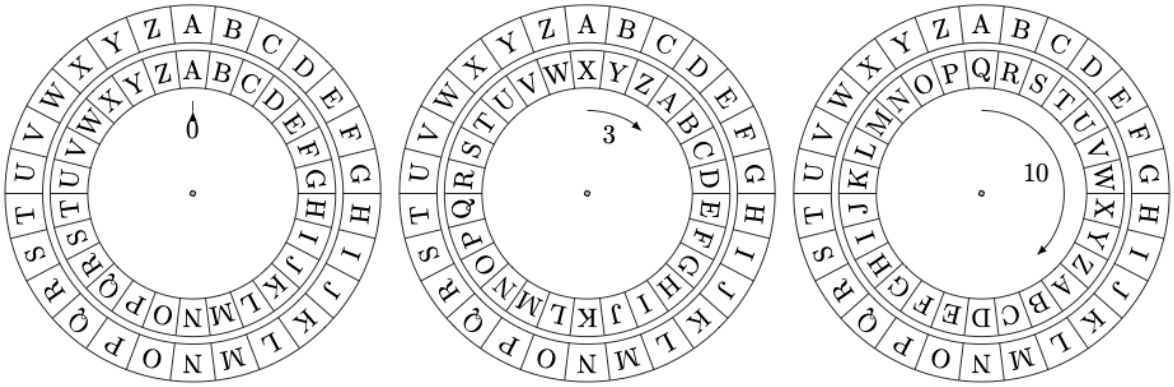
- יהי אלפבית Σ. תהי הפונקציה encode המקבלת 2 פרמטרים: מחרוזת לקידוד (הצפנה), שנסמנה - s ו- $k \in \mathbb{N} \cup \{0\}$. encode מצפינה את s על ידי כך שעבור כל $c \in s$ המקיים $c \in \Sigma$ היא מבצעת k הזחות ציקליות ימינה.
  כשאומרים שהפעולה ציקלית, הכוונה היא לכך שהפעולה היא מעגלית – ולכן למשל, אם $\Sigma = \{'a','b','c'\}$ אזי הזחה ימינה של התו 'b' ב-2 מיקומים תחזיר את הערך 'a' לאור תכונת המעגליות.
  במילים אחרות, encode ״דוחפת״ ימינה k פעמים כל אות אלפביתית ב-s.
  לדוגמה: אם k=1, אזי $'A' \mapsto 'B', 'K' \mapsto 'L'$ בעוד אם k=2 אזי $'A' \mapsto 'C', 'K' \mapsto 'M'$.

- יהי אלפבית Σ . תהי הפונקציה decode המקבלת 2 פרמטרים: מחרוזת לפענוח, שנסמנה - s ו- $k \in \mathbb{N} \cup \{0\}$. decode מפענחת את s על ידי כך שעבור כל $c \in s$ המקיים $c \in \Sigma$ היא מבצעת k הזחות ציקליות שמאלה.
  במילים אחרות, decode "דוחפת" שמאלה k פעמים כל אות אלפביתית ב-s.
  לדוגמה: אם k=1, אזי $'L' \mapsto 'K'$, $'A' \mapsto 'B'$ בעוד אם k=2 אזי $'M' \mapsto 'K'$, $'A' \mapsto 'C'$.

כפועל יוצא מההגדרות הנ"ל, נניח כי תהי s מחרוזת ו- $k \in \mathbb{N} \cup \{0\}$ ערך היסט, אזי נקבל

$$s = decode(encode(s))$$

לסיום, בתקווה שהדבר יפשט את הדברים, שימו לב לאילוסטרציה הבאה:



3 התוכנה cipher

בתרגיל זה נממש את התוכנה cipher המאפשרת להצפין ולפענח קטעי טקסט באמצעות צופן קיסר.

3.1 קלט

התוכנית תקבל דרך ה (Command Line Interface) CLI- ארבעה ארגומנטים:

- Command- הפקודה שרוצים לבצע. הערך יהיה מסוג מחרוזת, כאשר ערכי המחרוזת החוקיים יהיו רק "encode" ו"decode"- (עוד על בדיקות תקינות, בהמשך).
- k- מספר ההזחות המבוקש (להצפנה/לפיענוח), כך ש- $k \in \mathbb{N} \cup \{0\}$

נזכיר שניתן להגדיר את צופן קיסר עם ערכי k שליליים. אנחנו בחרנו, כדי להקל עליכם, שלא לעשות זאת.

- נתיב לקובץ קלט – בקובץ זה יהיה את הטקסט שהמשתמש מבקש להצפין או לפענח.
- נתיב לקובץ פלט – אל קובץ זה נכתוב את הטקסט לאחר הביצוע של ההצפנה או הפיענוח.

3.1.1 קריאת הקלט ובדיקות תקינות

שימו לב לנקודות הבאות הנוגעות לקריאת הקלט:

- נזכיר שתוכלו לגשת לארגומנטים שהתקבלו מה CLI-באמצעות argc, argv.
- לא ניתן לבצע השוואה בין מחרוזות באמצעות אופרטור ההשוואה (כלומר ==). כדי לבצע השוואה, תוכלו להשתמש בפונקציה המובנית strcmp . שימו לב שכדי להשתמש בפונקציה זו עליכם לכלול בראש התוכנית שלכם את הפקודה #include <string.h>.
- בתרגיל זה, באופן חד פעמי, נתיר את השימוש בפונקציה atoi על מנת להמיר מחרוזות למספר.[1]

כמו כן, שימו לב להנחות הבאות על הקלט:

- **אינכם רשאים** להניח כי כמות הפרמטרים שתקבלו תקינה (כלומר שלא קיבלתם פחות ארגומנטים מהנדרש, או לחלופין – יותר ארגומנטים מהנדרש).
- **אינכם רשאים** להניח כי הפקודה שקיבלתם אכן חוקית.
- **אתם רשאים** להניח כי k אכן יהיה מספר שלם. עם זאת אינכם יכולים להניח דבר על הערך שלו, מעבר לכך שהוא יכנס לטיפוס של int.
- **אינכם רשאים** להניח דבר על הטקסט שקיבלתם (דרך הנתיב לקובץ הקלט). בפרט, אינכם יכולים להניח כי הטקסט אינו כולל אותיות שאינן באלפבית' האנגלי, שהטקסט אינו ריק וכדומה.
- לא ניתן להניח שהנתיב שקיבלתם לקובץ **הפלט** הוא של קובץ קיים. אם הוא קיים- יש לדרוס את הקובץ הקודם. אם הוא לא קיים יש לייצר קובץ חדש (ובשני המקרים לכתוב לתוכו את הטקסט לאחר ההצפנה/הקידוד כמובן). רמז : חשבו, מהו מצב הפתיחה המתאים של הקובץ?

[1] שימוש יותר נכון, חכם ובטוח יהיה למשל עם הפונקציה – strtol וכך נצפה שתעבדו בפעמים הבאות (אלא אם נאמר אחרת.)

3.1.2 טיפול בשגיאות

במקרים של שגיאה, עליכם להדפיס את המחרוזת הרלבנטית מהרשימה שלהלן ל stderr-ולצאת באופן מידי מהתוכנית עם קוד שגיאה[2].

שימו לב שעליכם לוודא שאתם סוגרים את הקבצים הפתוחים לפני היציאה מהתוכנית!

- אם כמות הארגומנטים שסופקה לתוכנית אינה תקינה, עליכם להדפיס את המחרוזת:

```
"Usage: cipher <encode|decode> <k> <source path file> <output path file>\n"
```

- אם הפקודה שקיבלתם (ארגומנט הcommand) אינה תקינה, עליכם להדפיס את המחרוזת:

```
"The given command is invalid\n"
```

- אם ערך ה k-שקיבלתם אינו תקין (זאת אומרת k<0), עליכם להדפיס את המחרוזת:

```
"The given shifts value is invalid\n"
```

- אם יש בעיה עם הקובץ (קיבלתם נתיב לקובץ קלט שאינו קיים/פתיחת הקובץ נכשלה), עליכם להדפיס את הפקודה:

```
"The given file is invalid\n"
```

במידה ויש כמה שגיאות, ההודעה שצריכה להיות מודפסת לstderr תהיה לפי הסדר החשיבות הבא-

1. כמות ארגומנטים אינה תקינה.
2. K לא תקין.
3. פקודת command אינה תקינה.
4. בעיה עם נתיב/פתיחת הקובץ.

3.2 פלט

תוכנת ה cipher-שלנו תצפין ותפענח רק אותיות שהינן באלפבית האנגלי. כל אות שאינה באלפבית, תישמר כפי שהיא בפלט המוצפן. במילים אחרות, במינוחים שראינו לעיל, נגדיר $\Sigma = \{'A', 'B', \ldots, 'Z'\} \cup \{'a', 'b', \ldots, 'z'\}$.

עתה, בהנחה שלא היו שגיאות (כמפורט לעיל) התוכנה תפעל כך:

---

[2]. קוד שגיאה בC הוא ערך int ששונה מאפס. אתם יכולים להשתמש בEXIT_FAILURE זהו int קבוע שיושב בקובץ header בשם stdlib ונוהגים להחזיר אותו במקרה של שגיאה. (על מנת להשתמש בו יש להוסיף לתוכנית #include<stdlib.h>)

- אם הפקודה שהתקבלה היא encode: התוכנית תכתוב אל קובץ הפלט את ההצפנה של המחרוזת שהתקבלה, באמצעות האלגוריתם שהוצג לעיל באשר לפונקציה encode ואותה בלבד. (כלומר אין להדפיס לstdout דבר או לכתוב אל תוך קובץ הפלט תוכן נוסף).
- אם הפקודה שהתקבלה היא decode: התוכנית תכתוב אל קובץ הפלט את הפיענוח של המחרוזת שהתקבלה, באמצעות האלגוריתם שהוצג לעיל באשר לפונקציה decode ואותו בלבד. (כלומר אין להדפיס לstdout דבר או לכתוב אל תוך קובץ הפלט תוכן נוסף).

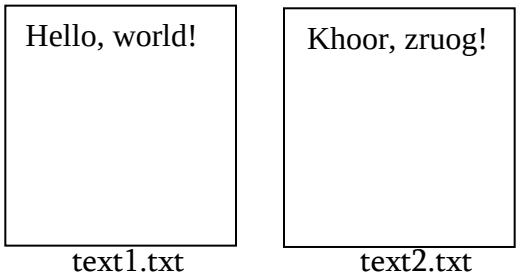
3.3 דגשים והנחיות נוספות:

- נדגיש שוב כי כל אות בטקסט שאינה מופיעה ב $\sum$ תודפס כפי שהיא.
- ראינו ש $\sum$ מורכבת מאותיות אנגליות גדולות וקטנות. שימו לב שהזחות ציקליות מתקיימות בנפרד בין האותיות הגדולות ובין האותיות הקטנות. זאת אומרת, לא יתכן שאות קטנה תהפוך לאות גדולה בעקבות הזחה ציקלית ולא יתכן שאות גדולה תהפוך לאות קטנה בעקבות הזחה ציקלית. (למשל, עבור k=1 $'z' \mapsto 'a', 'Z' \mapsto 'A'$.)
- זכרו כי פקודת המודולו (השארית) ב c המסומנת על ידי %, אינה תואמת לפקודת ה Modulo הנלמדת בשיעורי מתמטיקה.
- הנכם רשאים ליצור פונקציות עזר כראות עינכם.
- הנכם רשאים לעשות שימוש בספריה הסטנדרטית של C (למרות שניתן בהחלט לפתור את התרגיל עם שימוש בפונקציות strcmp ו atoi בלבד).
- זכרו להשתמש בקבועים ולהימנע מהשימוש במשתנים גלובאלים.
- אנו ממליצים להשתמש בפונקציות fgets ו- sscanf כדי לפרסר את קובץ הקלט.

4 דוגמה

נפתח בדוגמה המדגימה את האופן שבו התוכנית מקודדת את הטקסט "Hello, world!" שנמצא בקובץ text1.txt עבור פרמטר הסטה: k=3 וכותבת את הפלט אל הקובץ text2.txt.

```
$ ./cipher encode 3 text1.txt text2.txt
```



עתה, אם נרצה לפענח את הטקסט בקובץ text2.txt ("Khoor, zruog!") ולכתוב את הפיענוח אל text1.txt נוכל להריץ את התוכנית עם הארגומנטים הבאים:

```
$ ./cipher decode 3 text2.txt text1.txt
```

כאשר סימן ה-$ המופיע לעיל מסמן פקודה המבוצעת בשורת הפקודה (ב- Terminal).

5 נהלי הגשה

- קראו בקפידה את הוראות תרגיל זה ואת ההנחיות להגשת תרגילים שבאתר הקורס. כמו כן, זכרו כי התרגילים מוגשים ביחידים. אנו רואים העתקות בחומרה רבה!
- כתבו את כל ההודעות שבהוראות התרגיל בעצמכם. העתקת ההודעות מהקובץ עלולה להוסיף תווים מיותרים ולפגוע בבדיקה האוטומטית, המנקדת את עבודתכם.
- בשפת C יש פונקציות רבות שעשויות להקל על עבודתכם. לפני תחילת העבודה על התרגיל, מומלץ לחפש באינטרנט את הפונקציות המתאימות ביותר לפתרון התרגיל. ודאו שכל הפונקציות שבהן אתם משתמשים מתאימות לתקינה C99, וכי אתם יודעים כיצד הן מתנהגות בכל סיטואציה.

•פתרון בית הספר זמין בנתיב

~labcc/school_solution/ex1/cipher

- עליכם ליצור קובץ tar בשם "ex1.tar" (ובשם זה בלבד) הכולל אך ורק את הקובץ cipher.h . ניתן ליצור tar כדרוש על ידי:

```
$ tar -cvf ex1.tar cipher.c
```

- שימו לב: תרגילים שלא הוגשו בקובץ בפורמט tar או שיוגשו בשם השונה מ-"ex1.tar" לא יבדקו כלל ויקבלו ציון 0. נושא זה לא נבדק בPre-Submission Script.
- כדי להדר את התרגיל מהקובץ cipher.c לקובץ בינארי בשם cipher, תוכלו להשתמש בפקודה הבאה:

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

- אנא וודאו כי התרגיל שלכם עובר את ה Pre-submission Script ללא שגיאות או אזהרות. קובץ ה-Pre-submission Script זמין בנתיב

~labcc/presubmit/ex1/run <path_to_sumbission>

בהצלחה!!