



## המחלקה למדעי המחשב COMPUTER SCIENCE DEPARTMENT

סדנה מתקדמת בתכנות 61108  
סמסטר ב' תשפ"ד

### מטלה 2

#### מחרוזות, קבצים

### שאלה 1

כתבו פונקציה בשם `split`. הפונקציה תקבל אות קטנה `letter` ומחרוזת `str` המכילה מספר מילים (מילה היא רצף התווים שאינם רווח המופרד ממילים אחרות ע"י רווח אחד או יותר).

על הפונקציה לייצר מערך דינאמי של מחרוזות דינאמיות כך שכל מחרוזת היא מילה המופיעה ב-`str` ומתחילה באות `letter` או באות הגדולה התואמת ל-`letter` (המחרוזת שתהיה במיקום 0 במערך המחרוזות תהווה מילה ראשונה ב-`str` שמתחילה באות המתאימה, המחרוזת שתהיה במיקום 1 במערך המחרוזות תהווה מילה שנייה וכו', לפי סדר הופעתן ב-`str`).

הפונקציה תקצה את הזיכרון הנדרש למערך ולכל המחרוזות במערך המחרוזות, ולא יותר ממנו.

הפונקציה תחזיר את כתובת המערך, ותעביר (by reference) את גודלו. במידה והמערך לא מכיל אף איבר, כתובתו תהיה `NULL` וגודלו יהיה 0.

**לדוגמא**, עבור `str="Rony Goodman got a good mark"` ו-`letter='g'` הפונקציה תיצור מערך בעל שלוש מחרוזות הבאות:

```
Goodman  
got  
good
```

ניתן להניח שיש בזיכרון מספיק מקום להקצאה.

חתימת הפונקציה היא:

```
char **split(char letter, char *str, int *p_size);
```

## שאלה 2

### סעיף א'

כתבו פונקציה `void createFile(char *filename)` אשר מקבלת מחרוזת המכילה שם של קובץ טקסט. על הפונקציה ליצור קובץ בשם זה ולכתוב לתוך הקובץ רצף של תווים הנקלטים מהמקלדת.

### סעיף ב'

כתבו פונקציה `char commonestLetter(char* filename)` אשר תקרא קובץ טקסט ששמו מוכלל במחרוזת `filename`, ותחזיר את האות השכיחה ביותר בקובץ (האות המופיעה בקובץ הכי הרבה פעמים) מבין האותיות 'A' עד 'Z'. יש להתייחס אל אותיות קטנות וגדולות בטקסט באותו האופן (כך שגם 'a' וגם 'A' יחשבו כאות 'A'). על הפונקציה להחזיר את האות השכיחה בפורמט של אות גדולה.

אם יתגלו מספר אותיות עם אותה שכיחות מקסימלית, על הפונקציה להחזיר את האות הגדולה ביותר לפי הסדר הלקסיקוגרפי כלומר האות הקרובה ביותר לסוף בא"ב. במידה ואין אותיות בקובץ המתקבל כפרמטר או הקובץ לא נפתח, הפונקציה תחזיר תו אפסי ('\\0').

יש לבדוק (באופן המועדף אליכם) בשני הסעיפים האם קובץ נפתח בהצלחה.

### דוגמא:

עבור קובץ בעל התוכן

$d=AB+C-a/(b+c+a+bc)$   
 $D=2-a-b$   
 $c=6+d$

הפונקציה תחזיר 'C' (כל אחת מהאותיות 'A', 'B' ו-'C' מופיעות 4 פעמים אבל קוד ASCII של 'C' הוא גדול מקודי ASCII של 'A' ו-'B').

## שאלה 3

כתבו פונקציית פענוח `void decode(char *text)` המקבלת מחרוזת אשר מכילה טקסט מוצפן המורכב ממספר מילים.

על הפונקציה לפענח את הטקסט לפי השיטה הבאה:  
בכל מילה, יש להחסיר 1 מהערך ה-ASCII של התו הראשון, יש להחסיר 2 מהערך ה-ASCII של התו השני וכו'. את הרווחים יש להשאיר במקומם.

הפונקציה תחליף את התוכן המקורי של המחרוזת בתוכן החדש (שמהווה את הטקסט המפוענח).

לדוגמא, המחרוזת "bc8 d" תוחלף ב-"aa5 c".

## הוראות:

1. יש להשתמש כתבנית בקובץ Assignment\_2\_template.c המצורף אשר מכיל חלק מהקוד. צריך לשתול בתוכו את הטקסטים של כל הפונקציות הנדרשות במקום המתאים.
2. בהתאם לתבנית יש לאחד את כל השאלות הנ"ל לתוכנית אחת באמצעות תפריט הפונקציות Ex1(), Ex2(), Ex3() המשמשות להפעלת פונקציות השאלות 1,2,3 בהתאמה ומטפלות בהכנה ובקליטה של הנתונים המקוריים, בהדפסת התוצאות ובשחרור זיכרון דינאמי.
3. בפונקציה Ex1() למחרוזות המקוריות יש להשתמש במערך סטטי של תווים בגודל 100. יש להצהיר על משתנים (כולל המערך), לקלות את המחרוזת ואת האות, לקרוא לפונקציה split ולהציג את התוצאות. להדפסת מערך המחרוזות להשתמש בפונקציית עזר void printStringsArray(char \*\*str\_arr, int size) . בסוף יש לשחרר את הזיכרון באמצעות פונקציית עזר void freeMatrix (void \*\*A, int rows) .
4. בפונקציה Ex2() יש להצהיר על משתנים, ליצור קובץ בשם input.txt באמצעות פונקציה createFile, לקרוא לפונקציה commonestLetter עבור קובץ input.txt ולהציג את התוצאות.
5. בפונקציה Ex3() אין צורך בקלט המחרוזת המקורית. צריך לבחון פונקציה decode עבור הטקסט המוצפן הבא:

Btwlzx Dqges Eq|pj2 Tjlvqujs lqoqjy bpg Eqfxtx Xcwwtt

- יש להצהיר על המערך המאותחל, לקרוא לפונקציה decode ולהציג את הטקסט החדש.
6. אפשר להשתמש בפונקציות עזר נוספות לפי שיקול דעתכם. ניתן באמצעות #include להוסיף ספריות סטנדרטיות במידת הצורך.
  7. אין צורך בבדיקת תקינות הקלט ולא צריך לבדוק בתוך פונקציה את תקינות הפרמטרים שלה.
  8. יש להקפיד על ממשק ידידותי ככל האפשר.
  9. יש להשתמש בשמות משמעותיים ולתעד את הקוד עם הערות. יש להקפיד לכתוב בצורה מבנית.
  10. תכנית שלא עוברת קומפילציה לא תתקבל!