מבוא למדעי המחשב מ'/ח' (234114/7), סמסטר חורף 2017

תרגיל בית 4

מועד אחרון להגשה: **12.1 עד שעה 23:00**

המתרגל האחראי על תרגיל זה: **יותם אשל**
משרד: טאוב 315
E-mail: yotam.happy@gmail.com

שעות קבלה מיוחדות לשאלות על התרגיל(בנוסף על שעת הקבלה הרגילה) :
יום א' 1.1 בשעה 12:30, יום א' 8.1 בשעה 12:30
אם שעות אלו אינן נוחות לכם ניתן לתאם פגישה בשעות אחרות.

הנחיות:

- הגשה ב**בודדים**. עליכם לכתוב את הפתרונות לבד ולהגיש ביחידים.
- קראו את השאלות בעיון לפני שתתחילו בפתרונן.
- הקפידו לתעד את הקוד שלכם בהערות באנגלית.
- מלבד מילואים, לא יתקבלו תרגילים אחרי מועד הגשה. הגשה באיחור לאחר מועד הגשה נחשבת כאי-הגשה.
- כל יום מילואים = יום דחייה. על מנת לקבל את הדחייה, עליכם לשלוח באי-מייל, עותק של האישור המראה שהייתם במילואים (טופס 3010). אם האישור יגיע אליכם בתאריך מאוחר, יש להודיע על כך למתרגל האחראי על התרגיל לפני תאריך הגשת התרגיל.
- ערעורים ניתן להגיש עד שבוע לאחר קבלת הציון.
- **לא ניתן לערער על תוצאות הבדיקה האוטומטית.**
- **שימו לב! הבדיקה הינה בחלקה אוטומטית, ולכן הקפידו להדפיס בדיוק בפורמט שהתבקשתם ובידקו עם DiffMerge את הפלט שלכם מול הפלט של הדוגמאות שקיבלתם.**
  - השתמשו ב-redirection כדי להפנות את הפלט לקובץ טקסט.
  - וודאו את האותיות הגדולות והקטנות לפי הדוגמאות וההסברים בתרגיל.
  - **אין להדפיס רווחים שלא התבקשתם להדפיס (בתחילת שורה או בסופה).**
  - **בכל סוף שורה יש להדפיס תו ירידת שורה, גם בשורה האחרונה.**
  - השתמשו באתר הבדיקה העצמית.

- בתרגיל זה מותר להשתמש בפונקציות מהספריות stdio.h, stdbool.h **בלבד**, שנלמדו בהרצאות ובתרגולים.
- ההגשה הינה אלקטרונית ו**בבודדים** דרך אתר הקורס. קובץ ההגשה יהיה מסוג **zip** (ולא אף פורמט אחר) ויכיל בתוכו את הקבצים הבאים בלבד, ללא כל תיקיות:
  - קובץ **students.txt** עם מספר תעודת הזהות שלך וכתובת האי-מייל שלך.
  - קובץ פתרון **hw4q1.c** לשאלה 1
  - קובץ פתרון **hw4q2.c** לשאלה 2
  - קובץ פתרון **hw4q3.c** לשאלה 3
- **חובה לשמור את קוד אישור ההגשה שמקבלים מהמערכת לאחר שמגישים, עד לסיום הקורס.**
- יש להקפיד להגיש את כל הקבצים בדיוק עם השמות שמופיעים לעיל. הגשה שלא תעמוד בתנאי זה **לא תתקבל ע"י המערכת!** אם המערכת לא מקבלת את התרגיל שלכם, חפשו את הפתרון לבעיה באתר הקורס תחת הכפתור FAQ.

הנחיות לתכנון וכתיבת קוד: חשוב לקרוא לפני התרגיל(!)

גם בתרגיל זה חשוב להקפיד על תכנון הקוד **לפני** תחילת הכתיבה. לכן יש להקפיד על אותם הדרישות כמו בתרגיל 3 (אבל, אורך פונקציה יכול להיות עד 22 שורות).

1. אורך כל פונקציה לא יעלה על 22 שורות קוד (ראו הגדרות מדויקות בהמשך הדף). חשוב: עמידה בתנאי זה דורשת תכנון מוקדם של הקוד!
2. **רוחב כל שורה לא יעלה על 75 תווים**. ניתן לבדוק אורך של שורה בקודבלוקס (ראו תמונה מצורפת עם התרגיל). אם השורות ארוכות הן יגלשו בהדפסה, מה שיקשה על בדיקת התרגיל שלכם (ויגרור הורדת ניקוד), דרך נוספת הינה להגדיר כי יוצג קו המציין את קו 75 התווים settings ->editor -> margins and caret .
3. לפני כל פונקציה, יש לכתוב בהערה (בקצרה) מה הפונקציה עושה
   a. ההערה צריכה להסביר מה הפונקציה עושה ולא כיצד היא עושה זאת. לדוגמה: "הפונקציה מכניסה את x ו y למערך moves" אינו הסבר, לעומת "הפונקציה שומרת את המהלך האחרון במערך moves" הוא כן הסבר.

   או: "הפונקציה מקדמת את num אם isMove חיובי" אינו הסבר, לעומת "אם בוצע מהלך חוקי, הפונקציה מקדמת את מספר המהלכים החוקיים" (או בקצרה, "הפונקציה סופרת את מספר המהלכים החוקיים").
   b. ההערה צריכה להיות באנגלית. לצערנו עברית מודפסת כג"יבריש.
   c. ההערה צריכה להופיע לפני המימוש של הפונקציה ולא ההצהרה שלה.
4. חובה לתת שמות משמעותיים לפונקציות.
   a. השם צריך לשקף את מה שכתבתם בהערה (אך בתמצות).

בנוסף, יש להקפיד על הכללים הרגילים לגבי כתיבת קוד נכון:

5. חובה לתת שמות משמעותיים למשתנים שאתם מגדירים.
6. חובה להשתמש בהזחות תקינות כפי שנלמדו בתרגולים.
7. חובה להשתמש ב define במקומות המתאימים, כפי שנלמדו בתרגולים.
8. אסור להשתמש במשתנים גלובאליים או סטטים.
9. אסור לשכפל קוד שלא לצורך

הגדרות לגבי אורך פונקציה

1. מגבלת השורות תקפה לגבי כל פונקציה בקוד, **כולל ה-main**. יש לרשום את האורך בהערה לפני הפונקציה כדי להקל על הבדיקה.
2. אסור לכתוב 2 פקודות באותה שורה (למשל x=0; counter++;). לעומת זאת, אפשר להגדיר כמה משתנים באותה שורה ולאתחל אותם.
3. שורה המכילה הערות בלבד, כותרת הפונקציה ,סוגר '{', '}' אחד בלבד, או מילת else בלבד אינה נחשבת לספירת אורך הפונקציה.
4. כל שורה אחרת נחשבת לאורך הפונקציה, כולל: הגדרות משתנים, פקודת if, פקודת return וכל פקודה אחרת שאינה מופיעה ב 3.
5. שורה ארוכה שנשברה לשתיים (כדי שרוחב השורה יהיה קטן מ 75 תווים) נחשבת כ 2 שורות.

שאלה 1: עיבוד תמונות

תמונה במחשב מיוצגת כטבלה דו-מימדית (מטריצה) של ערכים. כל תא מכונה pixel (picture element). תמונה שמכילה בכל תא מספר שלם אי-שלילי מייצגת תמונת רמות אפור ("שחור-לבן"), ואילו עבור ייצוג תמונת צבע נדרשת עבור כל פיקסל שלישיית ערכים, אשר לעיתים תכופות מייצגים את רכיבי האדום, הירוק והכחול - אולם קיימים גם ייצוגים אחרים.

בשאלה זו נממש שיטה שימושית לביצוע חישובים מהירים בתמונות: טבלת שטח מסוכם. בהינתן תמונת רמות אפור בגודל $n \times m$ (n שורות על m עמודות), טבלת השטח המסוכם שלה מכילה בתא (i,j) את סכום כל התאים משמאלו ומעליו, וכן התא עצמו.

פורמלית, נסמן את התמונה המקורית כ- I, ואת טבלת השטח המסוכם כ-I_Σ אזי:

$$I_\Sigma[i][j] = \sum_{k=0}^{i}\sum_{l=0}^{j} I[k][l]$$

<u>הערה</u>: *בתמונות מקובל שהתא השמאלי העליון הינו $I[0][0]$, האינדקס הראשון מציין מס' שורה, והאינדקס השני – מס' עמודה, למשל ערך התא $I[2][0]$ בדוגמא הבאה הוא 4 – ערך התא הראשון בשורה השלישית.*

למשל, עבור התמונה הבאה:

$I =$

1	6	8	3
0	0	3	7
4	7	8	8
5	0	9	9

טבלת השטח המסוכם תהיה:

$I_\Sigma =$

1	7	15	18
1	7	18	28
5	18	37	55
10	23	51	78

לאחר שטבלת השטח המסוכם חושבה, נוכל להשתמש בה על מנת לקבל בחישוב פשוט סכום של ערכי הפיקסלים (בתמונה המקורית) בחלון מלבני מגודל כלשהו. עבור חלון הנתון ע"י האינדקסים

$$Rect = (i_{top}, j_{left}, i_{bottom}, j_{right})$$

הנוסחה לסכום ערכי הפיקסלים היא

$$\Sigma_{Rect} = I_\Sigma[i_{bottom}][j_{right}] - I_\Sigma[i_{top}-1][j_{right}] - I_\Sigma[i_{bottom}][j_{left}-1] + I_\Sigma[i_{top}-1][i_{left}-1]$$

למשל, הסכום של הריבוע המסומן בצהוב בדוגמא לעיל ($\mathrm{Rect} = (1,1,3,2)$) מחושב ע"י:

$$\Sigma_{Rect} = I_\Sigma[3][2] - I_\Sigma[0][2] - I_\Sigma[3][0] + I_\Sigma[0][0] = 51 - 15 - 10 + 1 = 27$$

חלק א'

ממשו פונקציה עם החתימה

```
void compute_integral_image(int image[][M], int n, int m, int integral_image[][M]);
```

הפונקציה מקבלת תמונת רמות אפור image *בגודל* $\mathrm{n} \times \mathrm{m}$ *(כמקודם* n *– מס' השורות,* m *– מס' העמודות) ומחשבת את טבלת השטח המסוכם עבורה לתוך* integral_image.

הניחו שאורך השורה המרבי הינו 50 פיקסלים, והגדירו את הקבוע M *בהתאם.*

*הניחו שהקלט תקין, ז"א ש-*image *ו-*integral_image *מצביעים חוקיים לזיכרון מוקצה בגודל המתאים, ו-*n *ו-*m *גדלים חוקיים שאינם חורגים מגודל הזיכרון המוקצה.*

***סיבוכיות**: על הפונקציה לעבוד בסיבוכיות זמן* $O(n \times m)$ *(לינארית בגודל התמונה) וסיבוכיות זיכרון נוסף* $O(1)$*.*
(רמז: מה הקשר בין ערך integral_image[i][j] *לבין* integral_image[i][j-1]*? לבין* integral_image[i-1][j]*?)*

חלק ב'

ממשו פונקציה עם החתימה

```
int sum_rect(int integral_image[][M], int rect[4]);
```

הפוקנציה תחשב את Σ_{rect} *לפי הנוסחה מעלה. גם כאן הניחו שהקלט תקין, בפרט* rect *מציין חלון מלבני בתמונה בדומה לדוגמה לעיל* $(i_{top}, j_{left}, i_{bottom}, j_{right})$

***סיבוכיות**: על הפונקציה לעבוד בסיבוכיות זמן* $O(1)$ *וגם סיבוכיות מקום נוסף* $O(1)$*.*

חלק ג'

ממשו פונקציה עם החתימה

```
void sliding_average(int image[][M], int n, int m, int h, int w, int average[][M]);
```

הפוקנציה מקבלת תמונה image *בגודל* $\mathrm{n} \times \mathrm{m}$ *ומחשבת לתוך* average *ממוצע על פני חלון נע בגודל* $h \times w$ *של הערכים בתמונה, כלומר, הערך בפיקסל* $average[i][j]$ *יהיה הערך הממוצע במלבן ב-*image *שמרכזו בפיקסל* (i, j)*, הגודל של המטריצה* average *כגודל תמונת הקלט.*

הניחו שמידות החלון אי-זוגיות (גם h *וגם* w*), וגם שהינן קטנות מגודל התמונה.*

עבור הערכים בקצוות (כאשר המלבן "לא נכנס" כולו בתמונה), הניחו שמחוץ לתמונה הערכים הם 0. למשל עבור מלבן 3×3 *והדוגמא לעיל,*

$$average[0][1] = (0 + 0 + 0 + 1 + 6 + 8 + 0 + 0 + 3)/9 = 2$$

<u>הערה</u>: כידוע, בפועל בשפת סי הערכים מחוץ למטריצה שיצרנו אינם בהכרח 0, עליכם לדאוג לכך במיוחד.

טבלת הערכים המלאה:

$$\text{average} = \begin{array}{|c|c|c|c|}\hline 1 & 2 & 3 & 2 \\ \hline 2 & 4 & 6 & 4 \\ \hline 2 & 4 & 6 & 5 \\ \hline 2 & 4 & 5 & 4 \\ \hline \end{array}$$

פורמלית, המטריצה average *מחושבת לפי:*

$$average[i][j] = \frac{1}{h \cdot w} \sum_{k=-\left\lfloor \frac{h}{2} \right\rfloor}^{\left\lfloor \frac{h}{2} \right\rfloor} \sum_{l=-\left\lfloor \frac{w}{2} \right\rfloor}^{\left\lfloor \frac{w}{2} \right\rfloor} image[i+k][j+l]$$

*(כאשר ההנחה שהערכים מחוץ ל-*image *הם 0).*

*שימו לב, מכיוון ש-*average *מכילה מספרים שלמים, את הממוצע יש לחשב בעיגול* ***למספר השלם הקרוב ביותר.***

סיבוכיות: *על הפונקציה לעבוד בסיבוכיות זמן* $O(n \times m)$ *וגם סיבוכיות מקום נוסף* $O(n \times m)$ *(שימו לב שלא צריכה להיות תלות בגודל החלון הנע).*

חלק ד'

כתבו תוכנית שקולטת מהמשתמש תמונה ומידות חלון נע, ומשתמשת בפונקציות מהסעיפים הקודמים על מנת לפלוט את טבלת השטח המסוכם וממוצע על פני חלון נע (תמונת ממוצעים). הניחו כי גודל תמונה לא יעלה על 50×50 *פיקסלים.*

דוגמא להרצת התוכנית:

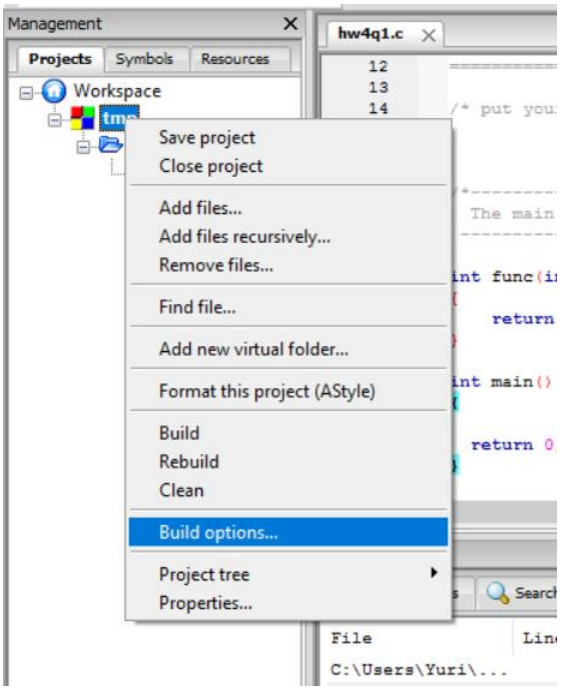
```
Enter image dimensions:
4 4
Enter image:
1 6 8 3
0 0 3 7
4 7 8 8
5 0 9 9
Enter sliding window dimensions:
3 3
Integral image is:
1 7 15 18
1 7 18 28
5 18 37 55
10 23 51 78
Smoothed image is:
1 2 3 2
2 4 6 4
2 4 6 5
2 4 5 4
```

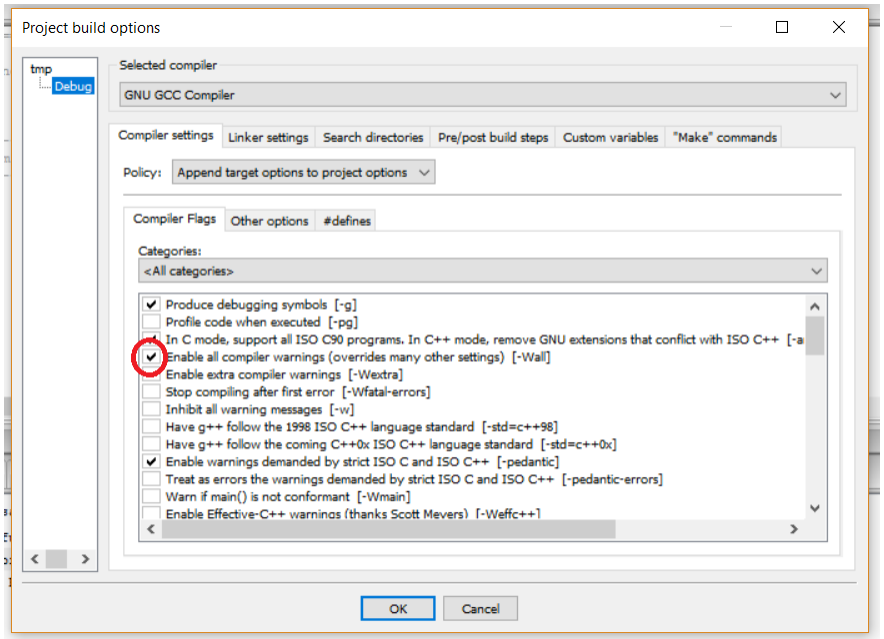
הנחיות לפתרון התרגיל

- *לאורך התרגיל הניחו שהקלט מהמשתמש חוקי, ז"א כאשר נדרש – המשתמש מזין ממדי תמונה תקינים* $(n, m \geq 1)$*, תמונה בגודל המתאים עם ערכים תקינים עבור הפיקסלים וכן מידות תקינות לחלון נע – ז"א מצבים לא תקינים לא יינתנו כמקרי קלט, ולכן הפלט המדויק במקרים אלו אינו משנה. יחד עם זאת,*

עליכם להקפיד על כללי תכנות נכון כפי שנלמדו בכיתה, לוודא תקינות קלט (למרות שלא ייבדק בבדיקה האוטומטית) במקרה של קלט לא תקין להתריע למשתמש ולסיים את התוכנית ע"י return מה-main, להקפיד על חלוקה הגיונית לפונקציות, שמות משמעותיים למשתנים ולפוקנציות, הגדרת קבועים וכו'.

- *מומלץ לפתור באמצעות top-down-design: התחילו את הפתרון מסעיף ד'. הניחו שגודל תמונה לא יעלה על 50x50 פיקסלים, והקצו את התמונה וטבלת השטח המסוכם כמטריצות דו-מימדיות בגודל המתאים. הגדירו פונקציות ריקות (או עם מימוש נאיבי) עם החתימות הנדרשות, וכן פונקציות נוספות לפי הצורך. כאשר אתם מגדירים פונקציות ריקות, על מנת שהמהדר לא יתלונן על משתנים לא בשימוש, אתם יכולים להוריד זמנית את הדגל Wall- דרך Build Options של הפרויקט (**לא לשכוח להחזיר את הדגל כאשר אתם בודקים!**):*





שאלה 2: סיווג מילים ע"פ סופית

חברת האח הגדול בע"מ רוצה לבנות מערכת אשר תאסוף מידע על משתמשים בפייסבוק: מה עשו, לאן הלכו ומה הם חושבים על זה. לשם כך הם רוצים לבנות מערכת אשר מסוגלת לקרוא טקסט, להבין את התוכן שלו ולנתחו. אחד השלבים הבסיסיים במערכת הינו כתיבת פונקציה אשר מזהה סוגי מילים בטקסט: שמות עצם, פעלים ותארים. מאחר ובאנגלית (כמו בשפות אחרות) ניתן במקרים רבים לזהות מילה לפי ה**סיומת** שלה (suffix), הוטל עליכם לבחון את השיטה הזו.

לצורך בחינת המערכת הוגדרו שלושה מילונים של סופיות, אחד לשמות עצם (nouns), אחד פעלים (verbs) ואחד לשמות תואר (adjectives):

Noun: ism, ics, er, ist, ment, ness

Verb: ate, ify, ize, ing, ed

Adjective: able, al, ful, ish, ive, ous

לנוחיותכם המילונים נשמרו בשלושה מערכים של מחרוזות, כאשר **כל סיומת כתובה מהסוף להתחלה וכל מילון ממוין בסדר לקסיקוגרפי** (יש להעתיק את הקבועים והמילונים למקום המתאים בתוכנית) :

```
#define N_NOUNS 6
#define N_VERBS 5
#define N_ADJS 6

.
.

int main(){
  char *noun_suffixes[] = {“msi”, “re”, “sci”, “ssen”, “tnem”, “tsi”};
  char *verb_suffixes[] = {“de”, ”eta”, “ezi”, “gni”, “yfi”};
  char *adj_suffixes[] = {“elba”, “evi”, “hsi”, “la”, “luf”, “suo”};
  .
  .
```

א. בשלב הראשון עליכם לממש פונקציה עם החתימה הבאה:

```
bool is_suffix_in_dict(char *str, char *dict[], int n);
```

הפונקציה מקבלת מילה (מחרוזת אשר מובטח כי אורכה אינו עולה על 32 תווים) ומקבלת מילון ומחזירה true אם המילה מסתיימת בסופית אשר נמצאת במילון. מאחר ועל המערכת לנתח כמויות גדולות של טקסט, על הפונקציה להיות יעילה ככל האפשר: בהינתן מילון באורך n עליה לרוץ בסיבוכיות זמן ריצה O(logn) וסיבוכיות זיכרון נוסף O(1). האורך המקסימלי של מילים וסיומות הוא 32 ולכן נחשב כקבוע לצורך חישובי סיבוכיות. להלן החוקים לפיהם יש להתאים מילים למילון:

a. שימו לב כי במילון אין סופית שבעצמה מסתיימת בסופית אחרת מהמילון. אם מצאתם סופית אפשר לסיים ואין צורך לבדוק האם יש עוד סופית ארוכה יותר שגם מתאימה.
b. יש להתעלם מתווים שאינם אות אנגלית לצורך הבדיקה. כלומר המחרוזת ”communism!“ מתאימה לסופית ”ism“. כמו כן ”communis,m“ מתאים גם הוא לסופית ”ism“.
c. על ההשוואה להיות case-insensitive, כלומר אין חשיבות לאות גדולה לעומת אות קטנה: CoMmUnIsM, COMMUNISM ו communism כולם מתאימים לסיומת ”ism“.

ב. עליכם לכתוב פונקציה עם החתימה הבאה:

```
bool read_sentence(char *noun_suffixes[], int noun_suffixes_len,
                   char *verb_suffixes[], int verb_suffixes_len,
                   char *adj_suffixes[], int adj_suffixes_len,
                   int *num_of_nouns, int *num_of_verbs, int *num_of_adjs);
```

הפונקציה מקבלת שלושה מילונים של סופיות, קוראת משפט מהקלט הסטנדרטי ומחזירה כמה מילים היו מכל סוג על פי הכללים הבאים:

a. ערך ההחזרה של הפונקציה יהיה true אם הגענו לסוף הקלט ו false אחרת.
b. את התוצאה יש לשים במשתנים המוצבעים ע"י `num_of_nouns, num_of_verbs, num_of_adjs`
c. אורך כל מילה לא יעלה על 32 תווים.
d. משפט הוא רצף של מילים מופרדות ביניהן ברווח מסוג כלשהו (רווח, טאב או \n).
e. משפט הוא רצף מילים מופרדות ביניהן ברווח כלשהו (רווח, טאב או ירידת שורה) המסתיים בתו נקודה ("I hate communism."). לא ניתן להניח מספר מקסימלי של מילים במשפט.
f. בהינתן משפט באורך k מילים, על הפונקציה לרוץ בסיבוכיות זמן O(klogn) וסיבוכיות מקום נוסף O(1).

ג. עליכם לכתוב תכנית אשר כותבת למשתמש "Enter text to analyze:\n", קוראת משפטים בזה אחר זה כל עוד לא מגיעים ל EOF ולאחר כל משפט מדפיסה את הנתונים שנאספו: "The sentence had __ nouns, __ verbs and __ adjectives.\n".

a. *ניתן להניח כי כל משפט מסתיים בנקודה, כולל האחרון.*
b. להלן מספר דוגמאות הרצה של התוכנית:

```
Enter text to analyze:
childish driver.
The sentence had 1 nouns, 0 verbs and 1 adjectives.
```

```
Enter text to analyze:
i am a DREAMER.
The sentence had 1 nouns, 0 verbs and 0 adjectives.
he is a pacifist...
The sentence had 1 nouns, 0 verbs and 0 adjectives.
```

```
Enter text to analyze:
zzzZZZ i am flying! flying! flying!.
The sentence had 0 nouns, 3 verbs and 0 adjectives.
10 hours to madness.
The sentence had 1 nouns, 0 verbs and 0 adjectives.
```

שאלה 3: סיבוכיות

בשאלות 1-6 בחרו את האות המייצגת את הסיבוכיות המתאימה מתוך טבלת הפונקציות להלן:

a. $\Theta(1)$
b. $\Theta(\log n)$
c. $\Theta(\log^2 n)$
d. $\Theta(\sqrt{n})$
e. $\Theta(\sqrt{n}\log n)$
f. $\Theta(\sqrt{n}\log^2 n)$
g. $\Theta(n)$
h. $\Theta(n\log n)$
i. $\Theta(n\log^2 n)$
j. $\Theta(n\cdot\sqrt{n})$
k. $\Theta(n^2)$
l. $\Theta(n^2\log n)$
m. $\Theta(n^2\log^2 n)$
n. $\Theta(n^2\cdot\sqrt{n})$
o. $\Theta(n^3)$
p. $\Theta(n^3\log n)$
q. $\Theta(n^3\log^2 n)$
r. $\Theta(2^n)$
s. $\Theta(n\cdot 2^n)$
t. $\Theta(n^2\cdot 2^n)$
u. $\Theta(n^3\cdot 2^n)$
v. $\Theta(3^n)$
w. $\Theta(n\cdot 3^n)$
x. $\Theta(n^2\cdot 3^n)$
y. $\Theta(n^3\cdot 3^n)$
z. *the correct answer doesn't appear*

```
void f1(int n){
        int s = 1;
        for(int i = 0; i < n; i++){
                s *= 2;
                for(int j = 0; j < s * n; j++){
                        printf("hi!\n");
                }
        }
}
```

1. מהי סיבוכיות הזמן של הפונקציה f1 כתלות ב-n?
2. מהי סיבוכיות המקום של הפונקציה f1 כתלות ב-n?

```
#define K 5

int f2(int n){
	int s=0;
	int d, m;
	while (n>1){
		scanf(“%d”, &m);
		d = (m > 2*K) ? 2*K : (m < K ? K : m);
		n = n / d;
		s += n;
	}
	return s;
}
```

3. מהי סיבוכיות הזמן של הפונקציה f2 כתלות ב-n?
4. מהי סיבוכיות המקום של הפונקציה f2 כתלות ב-n?

```
int f3(int k)
{
	int s=k/3;
	for (int m=2; k>0; k/=m)
	{
		s = s + k;
	}

	return 1+7*s;
}


void f4(int n)
{
	for (int jj=0; jj<n*f3(n); ++jj)
		for (int ii=1; ii*ii<n; ii*=2)
			printf("Hello jj!\n");
}
```

5. מהי סיבוכיות הזמן של הפונקציה f4 כתלות ב-n?
6. מהי סיבוכיות המקום של הפונקציה f4 כתלות ב-n?

אופן הגשת התשובות בשאלה 3:

באתר הקורס מסופק לכם קובץ טקסט בשם hw4q3template.c אשר נראה כך:

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

int main()
{
    int question;
    scanf("%d",&question);
    switch (question)
    {
    case 1 :
        printf("The answer to question 1 is a\n");
        break;
    case 2 :
        printf("The answer to question 2 is a\n");
        break;
    case 3 :
        printf("The answer to question 3 is a\n");
        break;
    case 4 :
        printf("The answer to question 4 is a\n");
        break;
    case 5 :
        printf("The answer to question 5 is a\n");
        break;
    case 6 :
        printf("The answer to question 6 is a\n");
        break;
    default :
        printf("Error!\n");
        break;
    }
    return 0;
}
```

הקובץ כמובן מתאים למקרה שבו בכל השאלות בחרתם בתשובה "a". כל שעליכם לעשות הוא להוריד את הקובץ מאתר הקורס, לשנות את האות "a" לאות המתאימה לתשובה שלכם בכל שאלה, ולשמור את הקובץ מחדש בשם hw4q3.c. את הקובץ hw4q3.c יש להגיש בתוך קובץ הזיפ יחד עם קבצי הקוד של שתי השאלות הקודמות והקובץ students.txt.

בהצלחה!