

שאלה מספר 1

שאלה זו עוסקת במערכים דו־ממדיים ובסיבוכיות

הקדמה

תמונה במחשב מיוצגת כטבלה דו־ממדית (מטריצה) של ערכים. כל תא מכונה pixel (קיצור של 'picture element'). תמונה שמכילה בכל תא מספר שלם אי־שלילי מייצגת תמונת Grayscale ("שחור-לבן"), ואילו עבור ייצוג תמונת צבע נדרשת עבור כל פיקסל שלישיית ערכים, אשר לעיתים קרובות 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 grayscale בגודל $m \times n$ (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 , אז:

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

הנוסחה לא הייתה מדויקת ותוקנה

הערה: בתמונות מקובלת שהתא השמאלי העליון הינו $I[0][0]$, האינדקס הראשון מציין מספר שורה, והאינדקס השני מציין מספר 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 המסומם 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:

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

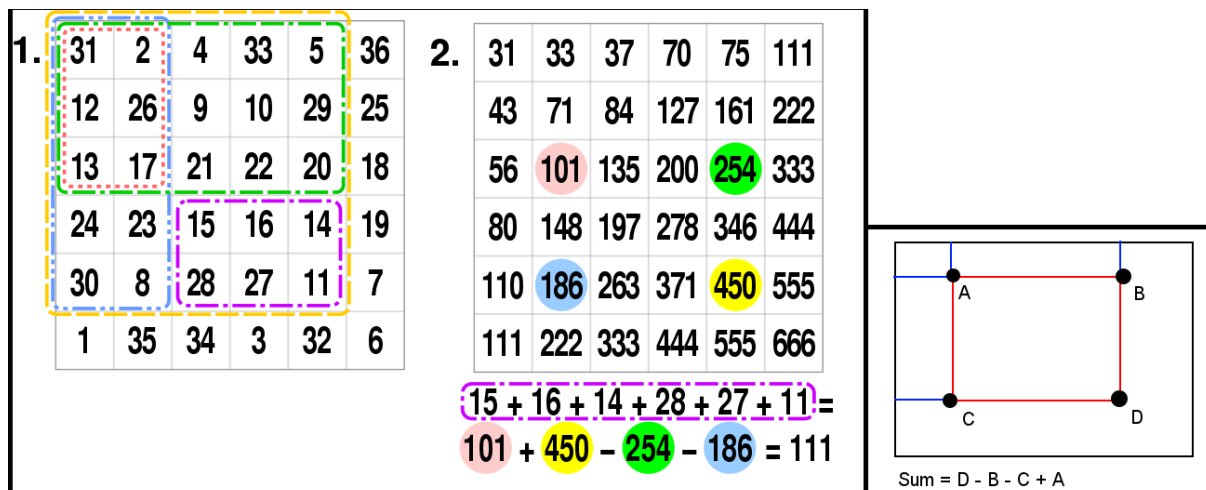
הנוסחה לסכום ערכי הפיקסלים היא:

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

למשל, הסכום של הריבוע המסומן בצהוב בדוגמא לעיל ($Rect = (1, 1, 3, 2)$) מחושב על ידי:

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

דוגמאות נוספות מ-Wikipedia על החישוב:



חלק א'

ממשו פונקציה עם החתימה:

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

הפונקציה מקבלת תמונת grayscale בגודל $n \times m$ (כמו קודם, n – מספר השורות, m – מספר העמודות) ומחשבת את טבלת השטח המסומם עבורה לתוך `integral_image`.

- הניחו שאורך השורה המירבי הינו 50 פיקסלים, והגדירו את הקבוע M בהתאם. באופן דומה, הניחו שאורך העמודה המירבי הינו 50 פיקסלים, והגדירו את הקבוע N בהתאם.
- הניחו שהקלט תקין, כלומר 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 $O(n \cdot m)$ (לינארית בגודל התמונה) וסיבוכיות זיכרון נוסף $O(1)$.

רמז למימוש: אם נוכל לחשב כל איבר במספר קבוע של פעולות ($O(1)$), נוכל לעמוד בדרישות הסיבוכיות (כי יש $n \cdot m$ איברים). לשם כך, חישבו מה הקשר בין האיבר ה- (i, j) ב-`integral_image` (אותו אנו רוצים לחשב בזמן קבוע) לבין האיברים ה- $(i-1, j)$, ה- $(i, j-1)$ וה- $(i-1, j-1)$ ב-`integral_image`.

חלק ב'

ממשו פונקציה עם החתימה:

```
#define RECT 4
int sum_rect(int integral_image[][M], int rect[RECT])
```

הפונקציה תחשב את Σ_{Rect} לפי הנוסחה מעלה. גם כאן הניחו שהקלט תקין, בפרט `rect` מציין חלון מלבני בתמונה בדומה לדוגמא לעיל:

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

- **סיבוכיות:** על הפונקציה לעבוד בסיבוכיות זמן $O(1)$ וסיבוכיות זיכרון נוסף $O(1)$.

חלק ג'

ממשו פונקציה עם החתימה:

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

הפונקציה מקבלת את טבלת השטח המסומם – `integral_image`, בגודל $m \times n$, ומחשבת לתוך `average` ממוצע על פני חלון נע בגודל $h \times w$ של הערכים בתמונה המקורית, כלומר הערך באיבר ה- (i, j) ב-`average` (כלומר `average[i][j]`) יהיה הערך הממוצע במלבן בתמונה המקורית שמרכזו בפיקסל (i, j) וגודלו $h \times w$.

- הגודל של המטריצה `average` כגודל תמונת הקלט.
- כמו כן, הניחו שמידות החלון אי-זוגיות (גם h וגם w), וגם שהינן קטנות מגודל התמונה.
- עבור הערכים בקצוות (כאשר המלבן "לא נכנס" כולו בתמונה. כלומר "גולש" מחוץ לתמונה), הניחו שמחוץ לתמונה הערכים הם 0 (מה זה אומר על הערכים בטבלת השטח המסומם שמחוץ לתחום?). למשל עבור מלבן 3×3 והדוגמא לעיל, מתקיים:

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

הערה: כידוע, בפועל בשפת סי הערכים מחוץ למטריצה שיצרנו אינם בהכרח 0, עליכם לדאוג למקרים אלה במיוחד.

טבלת הערכים המלאה עבור הדוגמה:

average =

1	2	3	2
2	4	6	4
2	4	6	5
2	4	5	4

פורמלית, המטריצה average מחושבת לפי (כאשר image היא התמונה המקורית):

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

כאשר שהערכים מחוץ ל-image הם 0.

שימו לב, מכיוון ש-average מכילה מספרים שלמים, את הממוצע יש לחשב בעיגול למספר השלם הקרוב ביותר (לשם כך יש להשתמש בפונקציה round בספריית math.h). שימו לב שכאשר אתם מחשבים את הממוצע אתם לא מבצעים חילוק שלמים בטעות.

- **סיבוכיות:** על הפונקציה לעבוד בסיבוכיות זמן $O(n \cdot m)$ וסיבוכיות זיכרון נוסף $O(1)$.
כאשר n ו- m הם גודל התמונה שקיבלנו. שימו לב שלא צריכה להיות תלות בגודל החלון הנע! כדי שנוכל לממש פונקציה זו בסיבוכיות הנדרשת, עלינו להיות מסוגלים לחשב כל איבר בזמן קבוע $O(1)$. לשם כך, עשו שימוש בפונקציה sum_rect, ב-integral_image ובנוסחה מההקדמה. שימו לב שיש לטפל במקרים של חריגה מגבולות התמונה.
- **רמז:** אנו רוצים לעשות שימוש ב-integral_image ובנוסחה מההקדמה. חישבו מה קורה כאשר חורגים מגבולות התמונה לשמאל ו/או למעלה ומה קורה כאשר גולשים לימין ו/או למטה. שני המקרים שונים.

הנחיות לפתרון התרגיל

- באתר הקורס מסופק לכם קובץ המכיל את פונקציית ה-main ופונקציית ההדפסה. יש להוריד את הקובץ ולממש את כל הפונקציות לעיל בקובץ זה.
- לאורך התרגיל הניחו שהקלט מהמשתמש חוקי, כלומר כאשר נדרש – המשתמש מזין ממדי תמונה תקינים ($n, m \geq 1$), תמונה בגודל המתאים עם ערכים תקינים עבור הפיקסלים וכן מידות תקינות לחלון נע – כלומר 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, להקפיד על חלוקה הגיונית לפונקציות, להקפיד על קריאות הקוד, שמות משמעותיים למשתנים ופונקציות, הגדרת קבועים וכו'.
- מומלץ לפתור באמצעות עקרונות top-down design. חלקו את העבודה לתתי-משימות, ובכל פעם ממשו תתי-משימה אחרת.

דוגמת הרצה:

```
Enter image dimensions:
4 5
Enter image:
1 2 3 4 5
6 7 8 9 10
11 12 13 14 15
16 17 18 19 20
Enter sliding window dimensions:
3 3
Integral image is:
1 3 6 10 15
7 16 27 40 55
18 39 63 90 120
34 72 114 160 210
Smoothed image is:
2 3 4 4 3
4 7 8 9 6
8 12 13 14 10
6 10 10 11 8
```

דוגמאות נוספות נמצאות בקבצי הקלט והפלט המסופקים לכם.

שאלה מספר 2

שאלה זו עוסקת במחרוזות ובהקצאת זיכרון דינאמית

בשאלה אתם מתבקשים לכתוב תוכנית המוסיפה לשמות עצם ביחיד בשפה האיטלקית את התוויות הבלתי-מיידעת.

הסטודנטים בקורס "איטלקית למתחילים 1" בטכניון (324630) למדו לאחרונה על התוויות הבלתי-מיידעת עבור שמות עצם בשפה האיטלקית, אך הם מתקשים בהבנת הכללים הדקדוקיים. לפיכך, הם פונים לסטודנטים בקורסים "מבוא למדעי המחשב מ' " (234114) ו-"מבוא למדעי המחשב ח' " (234117) על מנת שיכתבו עבורם כלי שיצרף את התוויות הבלתי-מיידעת עבור מילים בשפה זו.

בשפה האיטלקית קיימות 4 תוויות בלתי-מיידעות לשמות עצם ביחיד (un, una, un', un), והבחירה בתווית המתאימה תלויה במינו הדקדוקי של שם העצם ובאותיות הראשונות בו.

הגדרות בסיסיות

- **השפה האיטלקית** היא שפה רומאנית, שהיא שפה ממשפחת השפות ההודו-אירופאיות (זוהי משפחת השפות שבה נמצאת השפה האנגלית, אך לא משפחת השפות שבה נמצאות השפה העברית והשפה הערבית). האיטלקית הסטנדרטית היא הניב העיקרי המדובר כיום, ומקורו בניב שהיה מדובר בעיר פירנצה במאה ה-13. השפה האיטלקית מדוברת כיום בעיקר באיטליה, בארצות הברית, בצרפת, בגרמניה, בארגנטינה, בממלכה המאוחדת ובשווייץ.
- **שם עצם** (מוחשי) הוא מילה אשר מתארת עצם כלשהו. לכל שם עצם בשפה האיטלקית קיים מין דקדוקי.
- **תווית בלתי-מיידעת** היא מילה קצרה המצטרפת לשם עצם כדי להעיד על כך שהיא מתייחסת לעצם שאינו ידוע לשומע. בשפה העברית לא קיימת תווית בלתי-מיידעת, אך באנגלית קיימת – a / an (בשפה הערבית לא קיימת תווית בלתי-מיידעת, אם כי תופעת התנווין מקיימת זאת במידה מסוימת). באיטלקית, התווית הבלתי-מיידעת נמצאת לפני שם העצם (בדומה לאנגלית).
- **האלפבית האיטלקי** מורכב מ-21 אותיות מתוך האלפבית הלטיני, כולן מהוות חלק מהאלפבית של השפה האנגלית. באיטלקית אמנם קיימים סימנים דיאקריטיים על חלק מהאותיות, אך בשאלה זו נתעלם מהם.
- **אות תנועה** (vowel) באיטלקית היא אחת מבין האותיות הבאות:
 - a ○
 - e ○
 - i ○
 - o ○
 - u ○
 - h ○ כאשר היא האות הראשונה **במילה** (רק לעניין התווית הבלתי-מיידעת)
- **אות עיצורית** (consonant) היא אות שאינה אות תנועה.

• **אות שורקת** בשפה האיטלקית היא אחת מבין האותיות הבאות:

- z
- x
- y
- g שלאחריה n
- p שלאחריה s
- s שלאחריה אות עיצורית
- i שלאחריה אות תנועה

- **מין דקדוקי** של שמות עצם הוא מאפיין של שפה, לפיו קיימים 'שמות עצם זכריים' ו-'שמות עצם נקביים', וההתייחסות אליהן שונה. קיומו של מין דקדוקי הוא מאפיין של שפות אחדות (עברית, ערבית ואיטלקית לדוגמה), אך אינו קיים בכל השפות (אנגלית לדוגמה). עבור שמות עצם באיטלקית, המין הדקדוקי של שם עצם נקבע לפי האות האחרונה בו – o אם מינו הדקדוקי הוא זכר, ו-a אם מינו הדקדוקי הוא נקבה. באיטלקית קיימים שמות עצם נוספים ביחיד שאינם מסתיימים באותיות אלו, אך בשאלה זו נתעלם מהם.

כללי קביעת התווית הבלתי-מיידעת

בהינתן שם עצם כלשהו ביחיד, נוסיף לפניו את התווית הבלתי-מיידעת לפי הכללים הבאים:

- אם מינו הדקדוקי של שם העצם הינו זכר (מסתיים ב-o) והוא מתחיל באות שורקת, התווית הבלתי-מיידעת המתאימה לו היא uno. לדוגמה, uno xilofono.
- אם מינו הדקדוקי של שם העצם הינו זכר אך אינו מתחיל באות שורקת, התווית הבלתי-מיידעת המתאימה לו היא un. לדוגמה, un treno.
- אם מינו הדקדוקי של שם העצם הינו נקבה (מסתיים ב-a) והוא מתחיל באות תנועה, התווית הבלתי-מיידעת המתאימה לו היא un', ללא תו רווח בין התווית הבלתי-מיידעת לשם העצם. לדוגמה, un'amica.
- אם מינו הדקדוקי של שם העצם הינו נקבה אך אינו מתחיל באות תנועה, התווית הבלתי-מיידעת המתאימה לו היא una. לדוגמה, una donna.

חלק א'

ממשו את הפונקציה הבאה:

```
int get_gender(char* noun);
```

הפונקציה מקבלת את מילה באיטלקית שהיא שם עצם (מחרוזת), ומחזירה 1 אם מינו הדקדוקי של שם העצם הוא זכר ו-2 אם מינו הדקדוקי של שם העצם הוא נקבה. היעזרו בקבועים המוגדרים ב-define לצורך החזרת הערך.

לדוגמה, עבור ragazzo הפונקציה צריכה להחזיר את הערך 1, ועבור ragazza הפונקציה צריכה להחזיר את הערך 2.

חלק ב'

ממשו את הפונקציה הבאה:

```
int get_indefinite_article(char* noun);
```

הפונקציה מקבלת את מילה באיטלקית שהיא שם עצם (מחרוזת), ומחזירה מספר בהתאם לתוויות הבלתי-מיידעת המתאימה לשם העצם: 1 עבור un, 2 עבור una, 3 עבור una ו-4 עבור un. היעזרו בקבועים המוגדרים ב-define לצורך החזרת הערך. יש להשתמש בפונקציה שמומשה בחלק א' לטובת פתרון חלק זה.

לדוגמה, עבור treno הפונקציה צריכה להחזיר את הערך 1, עבור xilofono הפונקציה צריכה להחזיר את הערך 2, עבור donna הפונקציה צריכה להחזיר את הערך 3, ועבור amica הפונקציה צריכה להחזיר את הערך 4.

חלק ג'

ממשו את הפונקציה הבאה:

```
int get_length_indefinite_article(char* noun);
```

הפונקציה מקבלת את מילה באיטלקית שהיא שם עצם (מחרוזת), ומחזירה את אורך המחרוזת המתקבלת מהוספת התוויות הבלתי-מיידעת לפני שם העצם (כולל אורכו של שם העצם). יש להשתמש בפונקציה שמומשה בחלק ב' לטובת פתרון חלק זה.

לדוגמה, עבור amico הפונקציה צריכה להחזיר את הערך 8, שכן אורך המחרוזת "un amico" היא 8 תווים.

חלק ד'

ממשו את הפונקציה הבאה:

```
void add_indefinite_article(char* noun,  
char* noun_with_article);
```

הפונקציה מקבלת את מילה באיטלקית שהיא שם עצם (מחרוזת) וכתובת מערך של תווים בגודל מתאים שכל תאיו מאותחלים ל-0 (NULL), ומציבה במערך את התוויות הבלתי-מיידעת, לאחריה תו רווח (אם צריך) ולבסוף את שם העצם. התוצאה במערך צריכה להיות מחרוזת תקנית, כלומר עליה להסתיים בתו NULL. יש להשתמש בפונקציה שמומשה בחלק ב' לטובת פתרון חלק זה.

לדוגמה, עבור atleta הפונקציה צריכה להזין למערך שכתובתו נתונה את המחרוזת "un'atleta" (ללא תווי המירכאות העוטפים את המחרוזת).

חלק ה'

ממשו את הפונקציה הבאה:

```
void handle_indefinite_article(int max_length);
```

הפונקציה מקבלת חסם כלשהו מטיפוס `int`. הפונקציה קולטת (באמצעות `scanf`) מילה בודדת מהמשתמש, שהיא שם עצם באיטלקית אשר ניתן להניח שאורכה הוא לכל היותר כאורך החסם שהתקבל כקלט (לא כולל תו `NULL` בסוף), ומדפיסה את שם העצם שנקלט ולפניו התווית הבלתי-מיידעת המתאימה (ולבסוף תו ירידת שורה – `'\n'`). יש להשתמש בפונקציות שמומשו בחלק ג' ובחלק ד' לטובת פתרון חלק זה.

בפונקציה יש להקצות מערך אחד בגודל מינימלי עבור שם העצם, ומערך שני עבור הצירוף של התווית הבלתי-מיידעת ושל שם העצם – שני המערכים באמצעות הקצאה דינמית. יש לזכור לפנות את כל הזיכרון המוקצה באופן דינמי בתום השימוש בו.

לדוגמה, עבור הקלט 20 ועבור שם העצם `xilofono` הנקלט מ-`scanf`, יש להקצות מערך באורך 21 תאים עבור קליטת שם העצם ומערך באורך 13 תאים עבור המחרוזת המודפסת, ולהדפיס את המחרוזת "uno xilofono" (ללא תווי המירכאות העוטפים את המחרוזת).

הערות על התרגיל

- בתרגיל אין דרישה לסיבוכיות זמן ולסיבוכיות מקום. עם זאת, הקפידו על מימוש סביר (כל אחת מהפונקציות בתרגיל אינה אמורה להיות ארוכה מ-20 שורות).
- ניתן להניח שהפונקציה `malloc` מצליחה להקצות זיכרון בגודל המבוקש, שהפונקציה `free` מצליחה לפנות את הזיכרון שהוקצה באופן דינאמי, שהפונקציה `scanf` מצליחה לקבל קלט באופן תקין ושהפונקציה `printf` מדפיסה בהצלחה.
- יש לתעד את כל 5 הפונקציות בתרגיל. כמו כן, יש להסביר בהערה כל שורת קוד שאינה טריוויאלית.
- היעזרו בפונקציות הנמצאות בספרייה `string.h` – הן יקלו עליכם את המימוש.
- לא יבדקו מקרים שאינם מתוארים בתרגיל – מקרי הבדיקה אינם כוללים מחרוזות שהמבנה שלהם אינו תואם את המבנה של שמות עצם בשפה האיטלקית, או מחרוזות שאינן מציינות לכללים המתוארים.
- מקרי הבדיקה מכילים זוג מספרים (אחד מציין את האורך המקסימלי למילה והשני מציין את מספר המילים בקובץ הבדיקה), ולאחריהם מילים באיטלקית המופרדים במעברי שורה. פונקציית ה-`main` (הנתונה) קוראת את המספר הראשון, ומפעילה את הפונקציה אשר מומשה בחלק ה' (`handle_indefinite_article`) בלולאה.
- לא קיימים שמות עצם באיטלקית שאורכם הוא אות אחת – השתמשו בתכונה זו כדי לפשט את המימוש שלכם.

שאלה מספר 3

שאלה זו עוסקת בניתוח סיבוכיות של קוד

בשאלות הבאות בחרו את האות המייצגת את הסיבוכיות המתאימה מתוך הרשימה הבאה:

a. $\theta(1)$	n. $\theta(n^2 \cdot \sqrt{n})$
b. $\theta(\log n)$	o. $\theta(\sqrt{3^n})$
c. $\theta(\log^2 n)$	p. $\theta(n \cdot \log(\log n))$
d. $\theta(\sqrt{n})$	q. $\theta(n^3 \log^2 n)$
e. $\theta(\sqrt{n} \cdot \log n)$	r. $\theta(2^n)$
f. $\theta(\sqrt{n} \cdot \log^2 n)$	s. $\theta(n \cdot 2^n)$
g. $\theta(n)$	t. $\theta(n^2 \cdot 2^n)$
h. $\theta(n \cdot \log n)$	u. $\theta(n^3 \cdot 2^n)$
i. $\theta(n \cdot \log^2 n)$	v. $\theta(4^n \cdot \log n)$
j. $\theta(n \cdot \sqrt{n})$	w. $\theta(4^n \cdot \log^2 n)$
k. $\theta(n^2)$	x. $\theta(n \cdot 4^n)$
l. $\theta(n^2 \cdot \log n)$	y. $\theta(n^2 \cdot 4^n)$
m. $\theta(n^2 \cdot \log^2 n)$	z. the correct answer doesn't appear

קטע קוד מספר 1

חשבו את סיבוכיות הזמן והמקום של הפונקציה f1 (תנאי הביצוע של הלולאה שונה):

```
void f1(int n) {  
    int b = 2;  
    while(n > 1) {  
        b *= b;  
        n /= 2;  
    }  
    free(malloc(b * sizeof(int)));  
}
```

1. סיבוכיות זמן: _____

2. סיבוכיות מקום: _____

קטע קוד מספר 2

חשבו את סיבוכיות הזמן והמקום של הפונקציה f2:

```
void aux(int n) {
    while(n > 0) {
        printf("*");
        n -= 2;
    }
}

void f2(int n) {
    int k = n / 2;
    for(int i = 2; i < n; i *= i) {
        aux(k);
    }
    free(malloc(k));
}
```

3. סיבוכיות זמן: _____

4. סיבוכיות מקום: _____

קטע קוד מספר 3

חשבו את סיבוכיות הזמן והמקום של הפונקציה f3:

```
void f3(int n) {
    int x = 0;
    for(int i = 1; i < n; i *= 2) {
        for(int j = 1; j < n; j *= 3) {
            for(int k = 1; k < n; k++) {
                x++;
            }
        }
    }
}
```

5. סיבוכיות זמן: _____

6. סיבוכיות מקום: _____

בהצלחה!