לדמותן של שאלות במטלות ובבחינות בהוראת תכנות בעידן מודלי השפה הגדולים

כתב: ד״ר עפר אליאור, היחידה להוראת התכנות, האוניברסיטה העברית בירושלים

(א) מבוא

אם בשלהי 2022, עת נִתנה לראשונה לציבור הרחב גישה למודלי השפה הגדולים (Large Language Models; להלן: מודלים) הייתה שאלת השימוש בהם בהוראת התכנות עניין של ״האם״, הרי שבמהרה הפכה אצל רבים מאתנו המורים לשאלה של ״כיצד״. הדבר ניכר אפילו בסקירה שטחית של מבול הפרסומים שראו אור בשלוש השנים האחרונות בנוגע לתגובת עולם התכנות בכלל ועולם הוראת התכנות בפרט למודלים. תחילה בהדרגה, ולאחר מכן בקצב מהיר, הלכו והתקבלו בקרבנו המורים הן הכרה במגוון תועלות היכולות לצמוח משימוש במודלים בהוראה, הן אופטימיות ששימוש זה יאפשר להתאים טוב יותר את ההוראה ליכולותיהם של כל תלמידה ותלמיד. להכרה בצורך בשילוב המודלים בהוראת תכנות, ולמעבר להתמקדות בשאלת האופנים שבהם יש לבצע שילוב זה, היו ויש כמה ביטויים, ובהם: (א) פרסומם של מחקרים אקדמיים ומקורות מידע העוסקים בשימוש בכלי בינה מלאכותית לצורך פיתוח חומרי לימוד או מיטובם; (ב) יצירת כלים טכנולוגיים – בייחוד לומדות המבוססות על מודלים ועוזרי בינה מלאכותית (AI assistants)– המאפשרים לתלמידים שימוש מבוקר במודלים לצורך הבנת חומרי לימוד, פתרון תרגילים, ועוד; וגם (ג) הסתייעות במודלים לשם הפקת משובים והערכות לעבודת התלמידים (Denny et al., 2024).

בד בבד עם מגמות אלה, הולך ומתגבש מהלך נוסף ועקרוני בשילוב המודלים בהוראת התכנות. מהלך זה חותר לשלב את השימוש במודלים בתכנית הלימודים עצמה, הווה אומר, ללמד כיצד לתכנת באמצעות מודלים (Vadaparty et al., 2024; Cipriano and Ferreira, 2025). ביסוד מהלך זה מונחת תפיסה שלפיה על הוראת התכנות להכשיר תלמידים באופן מיטבי להשתלבות בעולם העבודה בפיתוח תכנה, ושממילא יש לקחת בחשבון את המגמות בתעשיית התוכנה בתפיסת מהותה של הוראת התכנות ובקביעת תכנית הלימודים.

השיקולים המדברים בעד השינוי והתומכים בשילוב הוראת תכנות באמצעות המודלים עולים מהיווכחות במגמות הללו. אציין שלושה מהם:

(א) שימוש במודלים הולך והופך לרכיב מהותי בעבודתם השוטפת של מרבית מפתחי התוכנה בתעשיה וגם בעבודתם של אחוז גדל והולך של חוקרים באקדמיה.

(ב) שלא כמו בשיעורי מבוא למדעי המחשב בכלל ושיעורי תכנות בפרט במסגרות חינוך, מפתחי תוכנה רבים היום אינם כותבים בעצמם תכניות מחשב מן היסוד, ובייחוד לא תכניות מחשב פשוטות, אלא משתמשים לצורך כך במודלים, ובעבודתם עוסקים בעיקר בקריאת קוד, בשכלולו ובשיפורו, וגם זאת בהתבססם במידה רבה על מודלים.

(ג) רבים מתלמידינו מסתייעים במודלים ללמידה, גם אם איננו מעודדים זאת. מכאן עולה הצורך לנכס את הלמידה העצמאית באמצעות מודלים, ולכל הפחות ללמד שימוש נכון ויעיל בהם.

השקפות פדגוגיות אלו נמצאות בראשיתו של תהליך יישום בארץ ובעולם. יש להניח שתהליך זה יילך ויתרחב. כבר היום ברי כי השלכותיו על הוראת התכנות המסורתית יהיו רבות אנפין ויובילו להוראת תכנות בדרך חדשה.

במאמר זה אדון בהבט אחד של תמורה חשובה זו, והוא השפעתה על **צביונן של שאלות במטלות בית ובבחינות בקורסי תכנות**. המאמר נחלק לשלושה חלקים. בחלק הראשון אדון בקווים מנחים חשובים בעיצוב אירועי הערכה בהוראת התכנות החדשה. בחלק השני אציג דוגמות לשאלות במטלות בקורס פייתון שעוצבו לאורם של קווים מנחים אלה. לבסוף אבחן מספר הבטים חשובים נוספים של כתיבת השאלות ובדיקתן.

(א) קווים מנחים בעיצוב מטלות ובחינות

בעיצוב מטלות ובחינות בהוראת תכנות המבוססת על התפיסה שתכנות הוא תכנות באמצעות מודלים, יש שני קווים מנחים עקרוניים.

קו מנחה אחד הוא הנחלת כישורי השימוש במודלים (Denny et al., 2024; Reeves et al., 2024). על השאלות באירועי הערכה לחתור לפתח אצל התלמידים כישורי כתיבה של מַנְחים (prompts)[1] יעילים; מנחה יעיל – היינו מנחה שהזנת תכנו למודל מביאה להפקת התשובה הנדרשת למזין המנחה. בנוסף, בכתיבת השאלות יש לחתור לקדם כישורי שימוש במודלים למגוון מטרות, ובהן: הערכת איכותם ויעילותם של פתרון לבעיה ושל האלגוריתם שהפתרון מבוסס עליו; איתור שגיאות בפתרון לבעיה, אי מילוי דרישות שהוצבו לפתרונה, או אי טיפול בקלטים בלתי צפויים של המשתמשים; שיפור פתרון לבעיה, הצעת פתרונות חילופיים, איתור הנחות סמויות של בעיה ומקרי קצה שלה, והשוואת פתרונות שונים זה מזה לבעיה אחת. כמו כן השאלות יטפחו עיון זהיר וביקורתי בתשובות המודלים, וכן הקפדה לוודא שאין אי דיוקים בתשובות, בייחוד בבעיות שהן מורכבות או תלויות בהקשרים בלתי שכיחים. נושאים נוספים שהשאלות יעסקו בהם: השוואת תשובות המודלים לתשובות המוצעות במקורות מסוגים אחרים, תוצאות בלתי רצויות של הסתמכות יתר על מודלים, בזבוז משאבים לתיקון קוד שגוי שיצר מודל, או בעיות בשימוש חוזר של קוד המוגן ברשיון.

הקו המנחה השני הוא קידום פיתוחם של כישורי חשיבה חישובית (Matobobo et al., 2026; Erez et al, 2023). חשיבה חישובית (Computational Thinking) היא גישה לפתרון בעיות שבבסיסה תהליך חשיבה שתכליתו העיקרית הוא ניסוח בעיה וניסוח פתרונה שיבוצע באמצעות מחשב (Wing, 2006). תהליך זה יכול לנקוט במגוון גישות לפתרון בעיות, ובהן חשיבה אינדוקטיבית וחשיבה דדוקטיבית, חשיבה לוגית, הפשטה, הכללה, ומידול, ניסוח בעיות, פתירת בעיות ופירוק בעיות לתת בעיות, תבניות אלגוריתמיות ותרשימי זרימה, תכנון פתרונות, בדיקתם והשוואה ביניהם, כתיבת פסאודו קוד, איתור שגיאות וניפוין, ולבסוף עיצוב הקלט, מבני הנתונים לשמירתו, והפלט. חשיבותם של כישורים בחשיבה חישובית ידועה זה מכבר, ולא מעט מחקרים הראו כי הצבת קידומן בתור מטרת ההוראה מגדילה את אחוזי ההצלחה בלימודי תכנות ומקטינה את אחוזי הנשירה מהם. בעידן המודלים יש לשאלות המטפחות כישורים בחשיבה חישובית חשיבות נוספת, בשל הסיכון להסתמכות יתר על המודלים בקרב תלמידינו – הסתמכות שעשויה לפגוע בפיתוח חשיבה חישובית. זאת ועוד: הנחלת כישורים בחשיבה חישובית יכולה גם לסייע לתלמידים בכתיבת מנחים יעילים.

לעיצוב השאלות וכתיבתן לאור קווים מנחים אלה יש השלכה מידית ובלתי נמנעת על מגוון השאלות שהתלמידים נשאלים לצורך תרגול התכנים הנלמדים. שאלות אלו אינן יכולות להיות אך ורק שאלות הבאות לבחון יכולת כתיבת קוד באופן עצמאי; בעידן המודלים יצטרפו להן שאלות מסוגים אחרים. עם אלו נמנות שאלות הבאות לבחון יכולת קריאת קוד, הבנתו והסברתו, וכמובן שאלות הבאות לבחון כישורים בשימושים במודלים וכישורים בחשיבה חישובית. שאלות קריאת קוד נהוגות בהוראת התכנות המסורתית,

[1] לפי האקדמיה ללשון העברית, מַנְחֶה הוא המונח העברי המקובל למונח prompt.

ולאחר הטמעת הוראת השימוש במודלים בתכנית הלימודים תהיה להן חשיבות נוספת, הואיל ומיומנות בקריאת קוד מסייעת בהערכת קוד שיוצרים מודלים.

(ב) דוגמאות לשאלות

להלן אדגים את השלכותיהם של קווים מנחים אלה על דמותן של שאלות הנשאלות בהוראת התכנות החדשה. הדוגמאות לקוחות מתרגילי בית בקורס מבוא לפייתון הנלמד באוניברסיטה העברית בשנה"ל תשפ"ו.

לקורס שני חלקים. בארבעת השיעורים הראשונים נלמדים נושאים יסודיים בתכנות, ובייחוד ערכים, משתנים, הוראות השמה, ביצוע מותנה, ביצוע חוזר וכתיבת פונקציות. בשיעורים אלה עדיין אין משולב שימוש במודלים בהוראה ובתרגול, וזאת כדי להפחית ככל הניתן פגיעה אפשרית ששימוש זה עלול להסב להבנת יסודות התכנות. שאלה 1 המוצגת כאן מופיעה בתרגיל בית מחלק זה של הקורס. בתרגילים אלה, כמו בכל תרגילי הבית בקורס, שובצו שאלות המעודדות פיתוח של כישורי חשיבה חישובית.

שאר השיעורים בקורס עוסקים בעיקרם במבני הנתונים הבסיסיים בשפה (מחרוזת, רשימה, רשומה, קבוצה ומילון) ובטיפול בקבצי טקסט. חומרי הלימוד של שיעורים אלה והתרגילים הנלווים להם מבוססים על שימוש במודלים והם חותרים לקדם את כישורי השימוש בהם. שאלות 2–4 מופיעות בתרגילי בית מחלק זה של הקורס. בכתיבת השאלות, ובייחוד שאלות שאינן מבוססות על שימוש במודלים, ניתנה הדעת לעודד את התלמידים להשיב באופן עצמאי ולהפחית ככל האפשר את היכולת להסתמך אך ורק על מודלים. הדבר נעשה בין השאר באמצעות ניסוחים עמומים או בלתי מפורטים. הסימן כוכבית לצד שאלה או סעיף של שאלה מופיע במקור, והוא בא לציין שאין להסתמך בפתרונה על שימוש במודלים. סימנים אלה הם ביטויים להיבט כללי יותר של הוראת התכנות בעידן המודלים, והוא השיתוף המלא של התלמידים במטרותיה של תכנית הלימודים החדשה. כחלק משיתוף זה יתקיים שיח פתוח בין המורים לתלמידים בצורך להימנע משימוש במודלים בשאלות מסוימות ובשלבים המוקדמים של הקורס.

שאלה 1* (מתרגיל בית 3, העוסק במבנה while)

(א) כתבו תכנית הקולטת בזה אחר זה מספרים שלמים עד לקליטת המחרוזת "stop". בסוף הקליטה יודפס המספר השלם הקטן ביותר שהוכנס.

(ב) שכללו את התכנית באופן שתדפיס הן את המספר הקטן ביותר הן את המספר שנקלט מיד לאחר מספר זה. אם המספר הקטן ביותר הוא המספר האחרון שנקלט הוציאו הודעה מתאימה. בפתרון שאלה זו חשוב במיוחד לא להסתמך על מודל שפה.

(ג) זהו את התבניות האלגוריתמיות שנלמדו בשיעור 3 ושעל פיהן בניתם את התכנית, ומדוע היה הכרח להשתמש בהן כדי לפתור את הבעיה.

בשיעור 3 נלמדות תבניות אלגוריתמיות בלולאה, ובהן: תבנית אמת/שקר, תבנית מניה, תבנית סכום, תבנית מכפלה, תבנית מינימום, תבנית מקסימום, ותבנית "הנסמך ל-" (איתור ערך באוסף ערכים המקיים תכונה מסויימת, והחזרת ערך אחר באוסף הקשור לערך שאותר – דוגמה: בקליטת זוגות שם-גיל, החזרת השם של הגיל המקסימלי).

סעיף א' מקדם מקדם כישורי פירוק בעיות וזיהוי תבניות אלגוריתמיות, ובפרט תבנית משתנה צובר לשמירת מצב (ערך מינימלי). התלמידים מחויבים להבין שמעקב אחר ערך מינימלי דורש שמירת מצב במהלך ביצוע הלולאה, תוך טיפול בקלט באורך בלתי ידוע. סעיף ב' הוא ליבת השאלה: הוא מחייב ניתוח מצבים מורכבים וטיפול בהם באמצעות מעקב אחר שני ערכים תלויים (המינימום והעוקב לו), תוך טיפול

במקרי קצה. דרישה זו מעודדת תרגול הפשטה של זרימת הקלט ופיתוח אסטרטגיה לזיהוי תלות זמנית בין איברים עוקבים. סעיף ג׳ מעודד עיון בחשיבה שעליה התבססו המענים לסעיפים א׳ וב׳ והתבססות על תבנית אלגוריתמית שנלמדה בכיתה ויישומה במודע על בעיה חדשה.

שאלה 2 (מתרגיל בית 5, העוסק במבוא לאוסקים)

(א) * עיינו בקוד זה וחזו מה יהיה הפלט שלו בלי להריצו. כתבו את התחזית שלכם בעברית והסבירו את ההגיון בה.

```
data = "programming"

result1 = sorted(data)

result2 = sorted(set(data))

print("Original:", data)

print("Result 1:", result1)

print("Result 2:", result2)

print("Lengths:", len(data), len(result1), len(result2))
```

(ב) כתבו מנחה המבקש ממודל שפה להסביר מה מבצע הקוד המוצג בסעיף א׳ צעד אחר צעד. על המנחה לבקש מהמודל לנתח כל שורה ושורה ולהסביר מדוע הערך ב-result1 שונה מהערך ב-result2.

(ג) הזינו את המנחה שכתבתם למודל והדביקו כאן את תשובתו המלאה של המודל. ציינו בראש הטקסט את המודל שהשתמשתם בו.

(ד) הריצו את הקוד מסעיף א׳ והשוו את התוצאה להסבר של המודל. כתבו בעברית הערכה להסבר שנתן המודל:

* האם ההסבר מדויק ושלם?

* האם הוא זיהה נכון את לב העניין (בקבוצה אין כפילויות)?

* כיצד התחזית שלכם בהשוואה לתוצאות בפועל מהרצת הקוד?

* האם הייתם משנים את המנחה שכתבתם כדי לקבל הסבר טוב יותר? אם כן – כיצד; אם לא – מדוע.

סעיף א׳ מעודד פיתוח של כישורי ניתוח קוד ומידול מנטלי (חיזוי ללא הרצה). המענה עליו מחייב הבנת הבדלים בין פעולות על מבני נתונים (במקרה זה, מיון רשימה לעומת הסרת כפילויות). סעיפים ב׳–ד׳ מטפחים כישורי כתיבת מנחים והערכה ביקורתית של פלט מודל. על התלמידים לנסח מנחה מדויק הכולל מונחים טכניים, להשוות את הסבר המודל לתוצאות הרצה בפועל, לזהות פערים, לאתר שגיאות, ולהציע שיפורים למנחה. הדגש על השוואה מפורשת בין התחזית העצמאית, הסבר המודל, והתוצאה האמתית, מעודד עיון בתהליך החשיבה ומחזק כישורי בדיקת פתרונות שיוצרים מודלים.

שאלה 3 (מתרגיל בית 6, בנושא בניית אוספים בלולאה)

(א)* כתבו את הפונקציה filter_with_extend. לפונקציה שני פרמטרים: numbers, רשימה לא ריקה של מספרים, ו-threshold, מספר. הפונקציה יוצרת רשימה חדשה שיש בה את כל המספרים ב-numbers הגדולים מ-threshold, ומחזירה רשימה זו. כדי לבנות את הרשימה החדשה הפונקציה משתמשת בפונקציה extend (על אף שעקרונית רצוי במקרה זה להשתמש דווקא ב-append; המטרה כאן היא תרגול הפונקציה extend).

(ב) בהתייעצות עם מודל שפה, בדקו את תקינות הפונקציה שכתבתם בסעיף א'. הקפידו לומר לו כבר במנחה הראשוני שלא יציג לכם את הגרסה התקינה של הפונקציה. כמו כן תנו לו דוגמה או שתיים לזימון הפונקציה. בקשו ממנו לבדוק היבטים מסוימים של הפונקציה, כגון תקינות הלוגיקה, האם השימוש בפונקציה extend נכון, האם הפונקציה מטפלת נכון במקרי קצה, וכן הלאה. הדביקו כאן את המנחה שהזנתם למודל ואת תיעוד השיחה המלא.

(ג) הסבירו בעברית את האופן שבו וידאתם את תקינות הפונקציה באמצעות המודל. אם המודל הציע תיקונים כתבו כאן את הקוד המשוכתב לפי תיקוניו.

המענה על סעיף א' מחייב תרגול בניית אלגוריתמים, תכנון פתרון (טיפול בפרמטרים), ושימוש מונחה במבני נתונים (רשימה) ובקרת זרימה (לולאות for) בשיטה מוגדרת (הפונקציה extend). סעיפים ב' וג' מתמקדים בכתיבת מנחים, ובקריאה ביקורתית של קוד ובתשובות מודל שפה. על התלמידים להשתמש במודל שפה ככלי לאימות, לאיתור שגיאות ולניפויין, כל זאת פי אילוצים ברורים: הם נדרשים לא לבקש מהמודל קוד מתוקן, לתת דוגמות מתאימות, ולנווט את המודל להתמקד בסוגיות מוגדרות (תקינות לוגית, שימוש נכון בפונקציה extend, טיפול במקרי קצה). לשם כך עליהם לכתוב מנחים יעילים וממוקדים ולתת את הדעת על מגבלות התשובות שמפיק מודל. בדרישה לנסח את שיטות האימות שנסמכו עליהן ולהצדיק תיקונים שנעשו, סעיף ג' מוסיף ומקדם הבנה אנליטית של תשובות המודל וגם מטפח מודעות בנוגע ליכולת להסתמך על סיוע שמעניק מודל.

שאלה 4 (מתרגיל בית 7, העוסק באינדקסים ברצפים וסריקת אינדקסים)

הפונקציה longest_identical_backward נועדה למצוא את הרצף הארוך ביותר של תווים זהים בסריקת מחרוזת לאחור. נתון מימוש שגוי של הפונקציה –

```
def longest_identical_backward(text):
    if len(text) <= 1:
        return len(text)
    max_length = 1
    current_length = 1
    for i in range(len(text) - 1, 0, -1):
        if text[i] == text[i - 1]:
            current_length += 1
        else:
            if current_length > max_length:
```

```
            max_length = current_length
        current_length = 1
    return max_length
```

(א)* בדקו את הפונקציה "ידנית" עבור שלושה זימונים שונים זה מזה שלה. זֶהו לפחות קלט אחד הגורם לפונקציה להתנהג באופן בלתי תקין. הסבירו את הבדיקות.

(ב) שוחחו עם מודל שפה לפי הצעדים האלה –

(1) הזינו מנחה זה –

Hello, I'm a student with no prior experience in programming, and I'm currently taking an introductory Python course. I have a few questions about Python and would appreciate it if you could tailor your responses to my beginner level.

(2) שתפו את המודל במימוש השגוי של הפונקציה ובקלט המדגים את ההתנהגות הבלתי תקינה. בקשו מהמודל שינתח את הבעיה.

(3) הציגו למודל שאלות המשך בבקשה להסברים נוספים להתנהגויות בלתי תקינות אחרות שקיבלתם. למשל –

Why might this loop miss the final sequence?

What happens at the boundary conditions?

(4)* במהלך השיחה תקנו במידת האפשר את הקוד השגוי.

בסיום השיחה הדביקו את התיעוד המלא שלה.

(ג) הדביקו את הקוד המתוקן שהתקבל בעקבות השיחה.

(ד) הסבירו בעברית כיצד תרמה השיחה לכם בתהליך ניפוי השגיאות מהקוד.

המענה על סעיף א׳ מעודד תכנון פתרון באמצעות מעקב ידני אחר ביצוע אלגוריתם, בדיקת בעיה באמצעות יצירת בדיקות (טסטים) וזיהוי מצבי קצה (בייחוד בעיית האיטרציה האחרונה), ותכנון לוגי תוך בחינת השאלה מדוע קלטים מסוימים מפיקים פלטים שגויים. סעיפים ב׳, ג׳ וד׳ מתווים פרוטוקול מובנה לשיחה עם מודל שפה. השיחה בהכרח פותחת בהצגת הֶקשרה (שיחה עם לומדי תכנות מתחילים). לאחר מכן התלמידים יוצרים מנחים שמטרתם לקבל סיוע בניתוח שגיאה קיימת בקוד ולא לבקש פתרונות. לבסוף הם מפנים למודל שאלות המשך אנליטיות בנוגע להתנהגויות הקוד במצבי גבול. השאלה כולה מוכוונת למנוע הסתמכות יתר ופסיביות ("העתק-הדבק" ממודל) ולעודד את התלמידים ככל הניתן לקחת לידיהם את המושכות של תהליך החשיבה: עליהם לתקן תיקונים בקוד בכוחות עצמם, ולאחר מכן להסביר כיצד השיחה עם המודל סייעה להם בניפוי שגיאות. כך הם לומדים להשתמש במודלי שפה ככלי אבחון החושפים פערים בחשיבה ובתכנון, וגם מפתחים מודעות בנוגע לתשובות המודל וליכולותיו. הדרישה לבדוק ידנית לפני השיחה עם המודל מבטיחה שחשיבה חישובית תנחה את ההסתייעות במודל.

(ג) היבטים נוספים של עיצוב השאלות וכתיבתן

נבחן כמה היבטים חשובים נוספים שיש לעיצוב השאלות בהוראת תכנות בעידן המודלים. האחד נוגע לבדיקת התרגילים. כפי שאפשר להיווכח מהדוגמות שהוצגו, הגדלת חלקן של שאלות שלפתרונותיהן נדרשים כישורי חשיבה חישובית וכישורי שימוש במודלים מובילה לריבוי בשאלות פתוחות (open-ended questions). שינוי זה מַקשה על בדיקה אוטומטית של עבודות התלמידים, כיוון שבדיקת שאלות טקסט חופשי היא עניין מסובך למערכות לבדיקות אוטומטיות של תרגילים. בעיה זו חשובה במיוחד בקורסים מרובי משתתפים. בספרות המחקר הוצע פתרון חלקי לבעיה, והוא נוגע לשאלות שהתשובות עליהן הן מנחים. לפי הפתרון המוצע, בדיקת התשובות תעשה באמצעות השוואת התשובה הנדרשת לפלט המתקבל מהזנת המנחה למודלים. במבט כללי יותר, חוקרים מתחילים לבדוק אם אפשר לקבוע באמצעות מודלים ציונים לשאלות שהמענה עליהן הוא בכתיבת טקסט חופשי, הן בהקשר הוראת התכנות הן בהקשרי הוראה אחרים. על אף שחקר הסוגיה הוא בראשיתו, כבר עלו ממנו מספר ממצאים חשובים. על פי אחד מהם, שיעור ההצלחה של מודלים בקביעת ציונים תקינים לשאלות מסוג זה דומה לשיעור הצלחתם של בודקים אנושיים ובמקרים מסוימים גבוה ממנו.

היבט חשוב אחר של עיצוב השאלות קשור בשמירה על יושר אקדמי. נושא זה מדאיג מורים רבים ומרתיע אותם מהכללת למידת השימוש במודלים בתכנית הלימודים. כאן יש לציין עיצוב השאלות באופנים מסוימים עשוי לסייע בהפחתת אי יושר מסוג זה. ראשית ריבוי בשאלות פתוחות עשוי לתרום לצמצומו, כיוון שקל יותר לזהותו בתשובות לשאלות מסוג זה. בנוסף רצוי לציין לצד כל שאלה במטלה אם במענה עליה אפשר להסתמך על מודלים אם לאו; סימון זה מסיר אי הבנות בקרב התלמידים. אפשר גם לדרוש מהתלמידים להסביר את החלטותיהם ושיקוליהם, או להעריך את תשובות המודל. בשאלות שפתרונן קוד אפשר לדרוש הוספת הערות לתרגילים המסבירות את קטעי הקוד ומציינות מה ממנו פרי עבודה עצמאית, או הגשת גרסות חלקיות שנוצרו במהלך העבודה על פתרונות לתרגיל והצדקות להחלטות מתודולוגיות שהתקבלו במהלך הפיתוח. ניתן גם להוריד את משקלם של ציוני מטלות שהשימוש במחשב בהן אינו מפוקח. מורים שירצו בכך יוכלו לנקוט בשיטות מסורתיות לוודא הוגנות, כגון ראיונות פתע בעל פה, בחינות ללא גישה לאינטרנט, וכו׳. כך או כך בהסתייעות במודלים נוכל לנסות ולברר באלו נושאים או באלו סוגי שאלות, או בתגובה לאלו ניסוחי מנחים, יתקשו מודלים להשיב כהלכה. אמנם בסופו של חשבון עלינו להניח כי תלמידים שירצו בכך יוכלו להגיש פתרונות מלאים שהפיקו מודלים בלי שנוכל לזהות זאת בוודאות.

לסיום נעיר הערות אחדות בנוגע לשאלות העוסקות בכתיבת מנחים. בכתיבת שאלות מסוג זה, ובכלל בעיצוב ההוראה החותרת לפתח כישורי כתיבה של מנחים, יש לבחון עד לאיזו מידה רצוי ללמד שיטות לעיצוב מנחים שהוצעו בשדה הידע והלמידה המקצועי ההולך ומתפתח – הנדסת מנחים (prompt engineering). כמו כן יש לתת את הדעת כי לכתיבה מיטבית של מנחים מועילה היכרות עם מילון מונחים מקצועיים, ובהם מונחים הנכללים כתבנית הלימודים, מונחים בהנדסת מנחים, ומונחים מהוראת התכנות וממחשוב בכלל, כגון אלגוריתם, קלט-פלט, פירוק בעיה, בדיקה, אימות וכן הלאה. לכן על מורים שעד כה נזהרו לדרוש מתלמידיהם זכירה של מונחים טכניים לבחון מחדש הימנעות זו. היבט זה של כתיבת המנחים קשור גם לשאלה אם במקומות שבהם שפת הלימוד אינה אנגלית, יתבקשו הסטודנטים לכתוב את המנחים ולנהל את שיחותיהם באנגלית, וכן לסוגיה הרחבה יותר של כישורי הכתיבה של תלמידים בשפה טבעית.

העיון בשאלת עיצובן של שאלות במטלות ובבחינות בעידן המודלים הוא בראשית דרכו. ככל שתתרחב התאמת הוראת התכנות לעידן זה כך תשתפר הבנת הסוגיה וככלל הבנתנו בנוגע למהלכים הנדרשים בצעידתנו לקראת הוראת התכנות החדשה

שלמי תודה

אני מודה לאורית חזן וליעל ארז על שיתוף הפעולה, העזרה והתמיכה בתהליך חשיבה שהוליד את המאמר. לוולנטין ונצק תודותיי שלוחות על שעיין בטיוטה מוקדמת של מאמר זה והעיר הערות חשובות.

רשימת מקורות

Cipriano, B.P. and Ferreira, L.S. 2025. Programmers aren't obsolete yet: A syllabus for teaching CS students to responsibly use large language models for code generation. arXiv abs/2502.15493 (2025). DOI: https://doi.org/10.48550/arXiv.2502.15493.

Denny, P., Prather, J., Becker, B. A., Finnie-Ansley, J., Hellas, A., Leinonen, J., Luxton-Reilly, A., Reeves, B. N., Santos, E. A., and Sarsa, S. 2024. Computing Education in the Era of Generative AI. *Communications of the ACM* 67, 2, 56–67. https://doi.org/10.1145/3624720

Erez, Y., Mike, K., and Hazzan, O. 2023. Leveraging computational thinking in the era of generative AI. *Communications of the ACM Blog*. https://cacm.acm.org/blogcacm/leveraging-computational-thinking-in-the-era-of-generative-ai/

Matobobo, C., Ncube, P.D.N., Ngesimani, N.L., Dzvapatsva, G.P., and Chinhamo, E. 2025. Enhancing computational thinking and problem-solving in programming education through generative AI: A scoped review. In Proceedings of the 2025 IEEE Global Engineering Education Conference (EDUCON). IEEE, London, United Kingdom, 1–9. DOI: https://doi.org/10.1109/EDUCON62633.2025.11016317.

Reeves, B. N., Prather, J., Denny, P., Leinonen, J., MacNeil, S., Becker, B. A., and Luxton-Reilly, A. 2024. Prompts First, Finally. *CoRR* abs/2407.09231. https://doi.org/10.48550/arXiv.2407.09231.

Vadaparty, A., Zingaro, D., Smith, D. H. IV, Padala, M., Alvarado, C., Gorson Benario, J., and Porter, L. 2024. CS1-LLM: Integrating LLMs into CS1 instruction. In *Proceedings of the 29th Annual ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '24)*. ACM, New York, NY, USA.

Wing, J. M. 2006. Computational Thinking. *Communications of the ACM* 49, 3, 33–35. https://doi.org/10.1145/1118178.1118215