מבוא למדעי המחשב מ'/ח' (234114 \ 234117)

מועד א' חורף תשפ"ה

מבחן מסכם , 28.02.25

מספר סטודנט: | | | | | | | | | *רשום/ה לקורס:* | 2 | 3 | 4 | 1 | 1 | |

משך המבחן: שלוש שעות.

הנחיות כלליות:

- בדקו שיש 8 עמודים (4 שאלות) במבחן, כולל עמוד זה.
- אלא אם כן נאמר אחרת בשאלות, **אין להשתמש בפונקציות ספריה או בפונקציות שמומשו בכיתה**, למעט פונקציות קלט/פלט והקצאת זיכרון (malloc, free). ניתן להשתמש בטיפוס bool המוגדר ב-stdbool.h.
- אין להשתמש במשתנים סטטיים וגלובליים אלא אם נדרשתם לכך מפורשות.
- הקפידו על סגנון כתיבה כזה שקורא התוכנית ידע לפענח את רעיונותיכם:
  a. התוכנית יכולה להכיל תיעוד קל להבנה.
  b. על התוכנית להיות כתובה באופן מסודר ומדורג.
  c. יש לתת שמות פונקציות ושמות משתנים משמעותיים.
  d. ערכים קבועים יש להגדיר בעזרת define .
  e. ניתן להוסיף פונקציות עזר כרצונכם.
- נוהל "לא יודע": אם תורידו את ההערה מהקריאה לפונקציה `printIDontKnow()` על שאלה שבה אתם נדרשים לקודד, תקבלו 20% מהניקוד. דבר זה מומלץ אם אתם יודעים שאתם לא יודעים את התשובה.
- שימו לב שקיבלתם את קבצי ה C ובהם שלד של התוכנית. עליכם רק להשלים את המימוש של הפונקציה הנדרשת.
- לכל שאלה קיבלתם מספר טסטים. מומלץ להוסיף טסטים שלכם.

המשך ההנחיות בעמוד הבא.

הנחיות הגשה: חשוב לקרוא בעיון!

1) לפני תחילת המבחן, יש למלא במודל הצהרה על טוהר הבחינות. סיסמה להצהרה במודל נמצאת בעמוד 3. **ניתן למלא את ההצהרה רק בחצי השעה הראשונה.**
2) הגשת כל שאלה במטלה נפרדת במערכת Gradescope לפי השלבים הבאים:
   1. ניתן להגיש מטלות רק לאחר ההצהרה על טוהר הבחינות.
   2. יש להתחבר ל- Gradescope אך ורק דרך הקישור המופיע באתר הקורס. כניסה שלא דרך המודל ולא אושרה על ידי צוות הקורס תגרור הליכים משמעתיים.
   3. **ההגשה תתאפשר החל מחצי שעה לאחר תחילת המבחן.**
3) יש למלא הצהרה על סיום הבחינה **לפני היציאה מהכיתה.**
4) במהלך המבחן מותר לעבוד אך ורק בסביבת CLion.
5) אסור השימוש בקבצים שנכתבו מראש בסביבת CLion, ניתן לפתוח אך ורק את קבצי השלד או קבצי טסטים.
6) אין להשתמש בשום כלי בינה מלאכותית יוצרת כגון: CoPilot, ChatGPT וכדומה. שימוש בכלים אלו יגרור הליכים משמעתיים.
7) הגשה לא לפי השלבים שפורטו לעיל, ובפרט: אי הצהרה על טוהר הבחינות/ הצהרה על סיום הבחינה, או הגשת מטלה לאחר הצהרה על סיום הבחינה, **תוביל לציון 0** והליכים משמעתיים.

נוסחאות שימושיות:

$$1+\frac{1}{2}+\frac{1}{3}+\ldots+\frac{1}{n}=\Theta(\log\ n) \qquad \log 1+\log 2+\ldots+\log n=\Theta(n\ \log\ n)$$

$$1^k+2^k+3^k+\ldots+n^k=\Theta(n^{k+1}) \qquad 1+k+k^2+k^3+\ldots+k^n=\Theta(k^n),\ k>1$$

סכום סדרה הנדסית: $S_n = a_1 * \frac{q^n-1}{q-1}$, אינסופית: $S_n = \frac{a_1}{1-q}$

סכום סדרה חשבונית: $S_n = n * \frac{a_1+a_n}{2}$

חוקי לוגריתמים:

$$\log_a(m\cdot n)=\log_a m+\log_a n$$
$$\log_a m^n = n\cdot\log_a m$$
$$\log_a a^x = x$$
$$a^{\log_a x} = x$$

צוות הקורס 234114/7

מרצים: גב' יעל ארז; פרופ׳ גרשון אלבר; מר איהב וותד **מתרגלים:** שלי גולן (מתרגלת אחראית), חן פרי, סער צור-שדי, גיא ארבל, שקד דרעי, אדיר רחמים, עומר צברי, אלעד לוי, מלק פרס, נוגה אפק, מתן ווינקלר, אופיר שבת, אורי דקל, צביקה לזר.

סיסמה להצהרה על טוהר הבחינות:

WinterLuck

שאלה 1: [25 נקודות]

א. חשבו את סיבוכיות הזמן והמקום של הפונקציה f1 המוגדרת בקטע הקוד הבא, כפונקציה של n. אין צורך לפרט את שיקוליכם. חובה לפשט את הביטוי ככל שניתן. ניתן להניח שסיבוכיות הזמן והמקום של printf היא $\Theta(1)$.

```
void f1(int n) {
    int m = n / 2;
    for (int i = 0; i * i < m; i++) {
        for (int j = i; j < m * 2; j++) {
            for (int k = 0; k < j - i; k++) {
                printf("%d ", i + j + k);
            }
        }
    }
}
```

ב. חשבו את סיבוכיות הזמן והמקום של הפונקציה f2 המוגדרת בקטע הקוד הבא, כפונקציה של n. אין צורך לפרט את שיקוליכם. חובה לפשט את הביטוי ככל שניתן. ניתן להניח שסיבוכיות הזמן של malloc ו- free הינה $\Theta(1)$.

```
void f2(int n) {
    int m = 0;
    for (int i = 0; i < n; i++) {
        m += i + 1;
    }
    int* arr = (int*)malloc(m * sizeof(int));
    if (arr == NULL) return;
    for (int i = 0; i < m; i++) {
        arr[i] = i * i;
    }
    for (int k = 0; k < m; k += 3) {
        int* b = (int*)malloc(k * sizeof(int));
        if (b == NULL) return;
    }
    free(arr);
}
```

ג. חשבו את סיבוכיות הזמן והמקום של הפונקציה f3 המוגדרת בקטע הקוד הבא, כפונקציה של n. אין צורך לפרט את שיקוליכם. חובה לפשט את הביטוי ככל שניתן. ניתן להניח ש- n>1.

```
int f3(int n) {
    if (n <= 1) {
        return n + 1;
    }
    int sum = 0;
    for (int i = 0; i * i < n; i++) {
        sum += i;
    }
    return sum + f3(n / 4) + f3(n / 4);
}
```

שאלה 2: [25 נקודות]

הגדרה: בהינתן מספר שלם וחיובי k, מערך מחרוזות יקרא "רכבת מחרוזות" אם קיים סידור כלשהו של **כל** המחרוזות במערך, כך שבין כל זוג מחרוזות סמוכות קיימת חפיפה של k תווים. כלומר, k התווים האחרונים במחרוזת ה- i-1 זהים ל-k התווים הראשונים במחרוזת ה- i.
"**שרשור רכבת מחרוזות**" הוא המחרוזת המתקבלת מכתיבת המחרוזות בסדר המתאים, כך ש-k התווים המשותפים בין המחרוזות נכתבים פעם אחת בלבד.

ממשו את הפונקציה:

```
void exam_q2(char** a, int n, int k, char* output);
```

המקבלת מערך מחרוזות a ממוין לפי k התווים הראשונים של המחרוזות a[1],…,a[n-1], אורכו n, מספר שלם חיובי k ומערך output אליו יש לכתוב את שרשור מערך המחרוזות הנתון.

קלט	תוכן המערך output בסיום התוכנית	הסבר
a = ["ZBC**DE**", "**AZ**BY", "**DE**X**FG**", "**FG**C**AZ**"] k=2	"ZBCDEXFGCAZBY"	כדי לבנות שרשור רכבת מחרוזות חוקי עלינו לשרשר את המחרוזת השלישית לראשונה. לאחר מכן, את המחרוזת הרביעית ולבסוף את השנייה. התווים המודגשים ב- bold הם אזורי השרשור המתאימים.

הערות:

- נתונה לכם תוכנית אשר קוראת מהמשתמש את המערך, אורכו, הערך k ומקצה מערך output בגודל מתאים.
- מובטח שקיים בדיוק שרשור אחד חוקי.
- מובטח ש-output מכיל מספיק מקום כדי לכתוב לתוכו את שרשור מערך המחרוזות הנתון באופן תקין כולל '\0'.
- מובטח כי המחרוזת הראשונה ב-a היא גם המחרוזת הראשונה בשרשור רכבת המחרוזות.
- שימו לב, המערך ממוין עבור האיברים a[1]…a[n-1], לא כולל האיבר הראשון.
- ניתן להניח שאורך כל מחרוזת במערך היא לכל היותר 30.

דרישות סיבוכיות:

- סיבוכיות זמן: $\Theta(k \cdot n \cdot logn)$.
- סיבוכיות מקום: $O(n)$.

שאלה 3 [25 נקודות]

ממשו את הפונקציה:

void exam_q3(int* arr, int n, int a, int b, int c)

המקבלת מערך ממוין בסדר עולה של מספרים שלמים arr בגודל n ושלושה מספרים שלמים a,b,c.
הפונקציה מממשת טרנספורמציה ריבועית על כל איבר במערך arr בהתאם לנוסחה:

$$a \cdot arr[i]^2 + b \cdot arr[i] + c$$

תוך כדי שמירה על המערך **<u>ממוין בסדר עולה</u>**.

<u>רמז:</u> פונקציה ריבועית בעלת נקודת אקסטרימום ב- $-\frac{b}{2a}$ כאשר $a \neq 0$. במקרה כזה, הפונקציה מונוטונית משני צידי נקודת האקסטרימום.

<u>לדוגמה (שימו לב למקרים השונים האפשריים):</u>

הסבר	המערך בסיום התוכנית	קלט
לאחר שינוי כל האיברים במערך: $arr[0] = 1 \cdot (-4)^2 - 4 \cdot 3 + 5 = 9$ $arr[1] = 1 \cdot (-2)^2 - 2 \cdot 3 + 5 = 3$ $arr[2] = 1 \cdot 2^2 + 2 \cdot 3 + 5 = 15$ $arr[3] = 1 \cdot 4^2 + 4 \cdot 3 + 5 = 33$ ולכן המערך הממוין בסיום התוכנית יהיה: arr=[3, 9, 15, 33]	[3, 9, 15, 33]	arr = [-4, -2, 2, 4] a=1, b=3, c=5
לאחר שינוי כל האיברים במערך: $arr[0] = -2 \cdot 2 + 1 = -3$ $arr[1] = -1 \cdot 2 + 1 = -1$ $arr[2] = 0 \cdot 2 + 1 = 1$ $arr[3] = 1 \cdot 2 + 1 = 3$ $arr[4] = 2 \cdot 2 + 1 = 5$ ולכן המערך הממוין בסיום התוכנית יהיה: arr=[-3, -1, 1, 3, 5]	[-3, -1, 1, 3, 5]	arr = [-2, -1, 0, 1, 2] a=0, b=2, c=1

<u>הערות:</u>

- נתונה לכם תוכנית אשר קוראת מהמשתמש את המערך, אורכו, ושלושת המספרים a,b,c. התוכנית מדפיסה את המערך arr בסיום ביצוע הפונקציה exam_q3 ועליו להיות ממוין.
- ניתן להניח שהקלט תקין ו- $a \geq 0$.
- ניתן להניח שהקלט לא יגרום ל- overflow ואפשר להשתמש במשתנים מטיפוס int בלבד.

<u>דרישות סיבוכיות:</u>

- סיבוכיות זמן: $\Theta(n)$.
- סיבוכיות מקום: $\Theta(n)$.

שאלה 4: [25 נקודות]

ממשו פונקציה:

int exam_q4(int* arr, int n, int x)

שמקבלת מערך של מספרים שלמים באורך n ומספר שלם x. מובטח שכל המספרים במערך הם בטווח בין $ARR_MIN = -100$ ל- $ARR_MAX = 100$. עליכם לחשב את מספר הזוגות במערך המניבים את הסכום x.
אם אין זוגות כאלו על הפונקציה להחזיר 0.

הסבר	פלט	קלט
קיים במערך שני זוגות שמניבים את הסכום 30: 100 + (-70) = 30 50 + (-20) = 30	2	arr = [-70, -20, 50 100] x=30
קיימים במערך ארבעה זוגות שמניבים את הסכום 0: arr[0] + arr[2] = 10 + (-10) = 0 arr[0] + arr[3] = 10 + (-10) = 0 arr[1] + arr[2] = 10 + (-10) = 0 arr[1] + arr[3] = 10 + (-10) = 0	4	arr = [10, 10, -10, -10, 30] x=0

<u>הערות:</u>

- נתונה לכם תוכנית אשר קוראת מהמשתמש את המערך, אורכו, והמספר x.
- ניתן להניח שהקלט תקין.
- כל זוג חייב להיות מורכב משני תאים זרים במערך.
- אסור להשתמש בלולאות.
- **<u>לא ניתן</u>** לקרוא לפונקציה exam_q4 מפונקציה אחרת מלבד ה- main.
- ניתן להשתמש בפונקציות עזר – **<u>הן חייבות להיות רקורסיביות בלבד.</u>**

<u>דרישות סיבוכיות:</u>

- סיבוכיות זמן: $\Theta(n)$.
- סיבוכיות מקום: $\Theta(n)$.

בהצלחה!