



## מבוא למדעי המחשב סמסטר א, מועד ב – תשע"ט

מרצים: ד"ר גלית חיים, ד"ר יהודה אלמליח

מתרגלים: מר חיים שפיר, מר דור חיים, גב' ענבל אברהם, מר אביב שוקרון

**קישור להורדת הבחינה:** <https://ck.cs.colman.ac.il/test.jsp>

### הקדמה

במבחן זה עליכם לענות על 4/4 שאלות תכנותיות ב C. משך המבחן 4 שעות. עליכם להקפיד היטב על ההוראות, ובפרט על הוראות ההגשה, שכן הבדיקה הינה אוטומטית (למעט השאלה על המבנים).

אתם מקבלים:

- ארבעה קובצי c וקובץ h אחד המכיל את כל החתימות הנדרשות.
- אין להוסיף include-ים כלשהם וגם אין בכך צורך.
- עליכם להשלים את הקוד בקובצי ה c בלבד, ע"פ הגדרות השאלה
- בנוסף, אתם מקבלים את קובץ ה mainTrain המכיל בדיקות לוגיות. זהו ה main של מוד האימון. הוא חשוף, וייתן לכם מושג לגבי הבדיקות הלוגיות של המבחן.
  - שימו לב שהוא אינו בודק את \*כל\* הבדיקות הלוגיות של המבחן, לכן עליכם להתרכז בהוראות המבחן ולא רק לגרום ל mainTrain להצליח.

עליכם להגיש:

- את קבצי ה c מושלמים. לא ב zip או דומין, אלא את קבצי המקור עצמם.

שימו לב: לא יתקבל כל ניקוד לשאלה שעבורה הקוד לא מתקמפל או שיש לו שגיאות בזמן ריצה

הבדלים בין מוד אימון להגשה:

מוד אימון	מוד הגשה
בדיקה חשופה המכילה חלק מהבדיקות	בדיקה חסויה המכילה את כל הבדיקות
ניתן להגיש אינספור פעמים עד לתום הבחינה	ניתן להגיש אינספור פעמים עד לתום הבחינה
משוב מלא – שגיאות קומפילציה, ריצה ולוגיות	משוב חלקי – שגיאות לוגיות בלבד
ציון מידי. זהו אינו ציון המבחן ולא חלק ממנו	ציון מידי – זהו ציון המבחן בגין שלושת השאלות הראשונות
לא נחשב כהגשת המבחן	לא נחשב כהגשת המבחן (הגשת המבחן תיעשה במוד "הגשה סופית")



---

שימו לב! רק שאלות 1-3 נבדקות אוטומטית, ואילו שאלה 4 נבדקת ידנית. לכן במערכת הבדיקות הציון המקסימלי שניתן לקבל הוא 60. ציון זה משמעותו קבלה של כל הנקודות על שאלות 1-3. לאחר הבדיקה הידנית של שאלה 4 יפורסם לכם ציון סופי למבחן.

---

עם תחילת המבחן ניתן יהיה להוריד את קבצי הבחינה ממערכת הבדיקות ויחולקו הטפסים. האינטרנט ינותק והמבחן יתחיל. אל תחכו לרגע האחרון להגיש את המבחן בלחץ, ואז לגלות ששכחתם משהו. תכננו את הזמן היטב.

**הגשה סופית:** לאחר הגשת הבחינה **(במוד הגשה סופית)** לא תוכלו להגיש יותר את הבחינה. מיד בתום ההגשה הסופית תקבלו מספר אסמכתא בעל 4 ספרות. העתיקו את המספר ומסרו אותו לבוחנת. בנוסף, עליכם ללחוץ על הקישור "לינק לשרת הגיבוי" ולהגיש את הפתרון שלכם גם בשרת הגיבוי.

קישור להורדת הבחינה: <https://ck.cs.colman.ac.il/test.jsp>

**חומר עזר:** אין חומר עזר

בהצלחה!



## שאלה 1: 20 נקודות

מטריצה הינה מערך דו-מימדי של מספרים. מטריצה מסדר  $n \times n$  נקראת מושלמת אם היא מכילה אך ורק את המספרים  $1, 2, \dots, n$  כך שכל מספר מופיע בדיוק פעם אחת בכל שורה ובכל עמודה.

לדוגמא :

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 4 & 3 \\ 3 & 4 & 2 & 1 \\ 4 & 3 & 1 & 2 \end{pmatrix} \text{ המטריצה היא מטריצה מושלמת אך}$$

$$\begin{pmatrix} 1 & 3 & 4 \\ 2 & 1 & 3 \\ 3 & 2 & 1 \end{pmatrix} \text{ המטריצה אינה מטריצה מושלמת, שכן, 4 גדול מ-} N \text{ (המטריצה בגודל } 3 \times 3 \text{)}$$

$$\begin{pmatrix} 1 & 3 & 3 \\ 2 & 1 & 3 \\ 3 & 2 & 1 \end{pmatrix} \text{ המטריצה אינה מטריצה מושלמת, שכן, בשורה ה-1 ובעמודה ה-3 הספרה 3 מופיעה פעמיים.}$$

כתבו פונקציה המקבלת מטריצה של מספרים שלמים (int) ומחזירה 1 אם המטריצה מושלמת ו-0 אחרת.

- גודל המערך הינו N על N ו N הוגדר ע"י #define.

להלן הצהרת הפונקציה :

```
int IsPerfect(int mat[][N]);
```

ממשו את הקוד הנדרש בקובץ Q1.c



## שאלה 2: 20 נקודות

כתבו פונקציה

```
void create_substitutor(char *keyword, char *subst)
```

המקבלת מחרוזת keyword המכילה רק אותיות גדולות באנגלית, ובה כל אות אנגלית מופיעה לכל היותר פעם אחת, ויוצרת מחרוזת subst שראשיתה היא keyword, והמשכה הוא שאר אותיות הא"ב האנגלי שאינן מופיעות ב-keyword, לפי הסדר שלהן בא"ב האנגלי.

לדוגמא, בהינתן:

keyword

S	T	A	D	I	U	M	\0												
---	---	---	---	---	---	---	----	--	--	--	--	--	--	--	--	--	--	--	--

לאחר הרצת הפונקציה המחרוזת subst תיראה כך:

subst

S	T	A	D	I	U	M	B	C	E	F	G	H	J	K	L	N	O	P	Q	R	V	W	X	Y	Z	\0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	----

הנחיות:

מותר להניח שהקלט המתקבל הוא חוקי (כלומר keyword מכילה אך ורק אותיות גדולות באנגלית, ואף אות לא מופיעה יותר מפעם אחת).

ממשו את הקוד הנדרש בקובץ Q2.c



## שאלה 3: 20 נקודות

שאלה זו עוסקת ברקורסיה.

**אין להשתמש בלולאות על מנת לפתור אותה.**

שני מספרים יוגדרו כקרובים אם **שניהם חיוביים שלמים** ומקיימים את התנאים הבאים:

1. יש להם אותו מספר ספרות.
2. ההפרש בין כל שתי ספרות בעלות אותו משקל (ספרת האחדות, ספרת העשרות, ספרת המאות וכו') הוא לכל היותר 1.

דוגמאות:

- המספרים 12345678 ו 13456788 הם מספרים קרובים.
- המספרים 123 ו 1230 הם לא מספרים קרובים כי אין להם אותו מספר ספרות.
- המספרים 123 ו -122 הם לא מספרים קרובים כי הם שליליים.
- המספרים 33 ו 84 הם לא מספרים קרובים כי ההפרש בין שתי הספרות של העשרות (3 ו 8) גדול מ 1.

כתבו פונקציה רקורסיבית שההצהרה שלה היא:

```
int areClose(long n1, long n2);
```

אשר מחזירה 1 אם המספרים  $n1$  ו  $n2$  הם קרובים ו 0 אחרת.

את הפתרון לשאלה זו יש לכתוב בקובץ Q3.c



## שאלה 4(40 נק')



יומן גוגל (Google Calendar נקרא בעבר בשם הקוד CL2) הוא יומן אירועים על בסיס לוח שנה ברשת, שמציעה חברת גוגל. היומן של גוגל יכול לשמש לצורך ארגון זמן. באמצעות כלי זה יכול המשתמש לקבוע פגישות, אירועים ותזכורות, לשתף את היומן ולקבוע דרגות אבטחה ביחס לשיתוף, ולהוסיף את חגי המדינה שיחפוף. כמו כן ישנה אפשרות לייצא ולייבא את נתוני היומן. השירות הוכרז ב-13 באפריל, 2006 ויצא משלב הבטא ביולי 2009. משתמשים אינם צריכים חשבון ג'ימייל כדי להשתמש בשירות, אבל הם נדרשים להיות בעלי חשבון בגוגל.

במסגרת שאלה זו תצטרכו לממש חלק מן היכולות של Google Calendar כמו גם יכולות שאינן קיימות עדיין ב-Google Calendar. מימוש הפונקציה יעשה בקובץ Q4.c

להלן ארבעה מבנים מרכזיים:

**תאריך (Date):** תאריך לקביעת פגישה. התאריך בנוי מיום (מספר בין 1 ל-31), חודש (מספר בין 1 ל-12), שנה, שעה (מספר בין 0 ל-23), דקות (מספר בין 0 ל-59). בהמשך תוכלו לראות כי תאריך יקושר למועד פגישה. אין צורך לבדוק חוקיות תאריכים פרט לטווחי המספרים שצוינו כאן – למשל, על אף שחודש פברואר לעולם לא יכלול 31 ימים, לצורך פשטות (משום שאין עניין זה מהות השאלה) ניתן להתייחס אליו בתרגיל כ-31 ימים.

**איש קשר (Contact):**

- name: שם איש הקשר (טקסט לא יותר מ-TEXT\_SIZE תווים)
- phone: מספר טלפון (טקסט לא יותר מ-TEXT\_SIZE תווים)
- emailAddress: כתובת אימייל (טקסט לא יותר מ-TEXT\_SIZE תווים)

**ארוע/פגישה (Event):**

- title: נושא הפגישה (טקסט לא יותר מ-TEXT\_SIZE תווים)
- startDate: תאריך (כולל שעה) לתחילת הפגישה/אירוע



- endDat: תאריך (כולל שעה) לסיום הפגישה/אירוע
- cont: מצביע למערך של מצביעים ל-Contact. אלו ה-Contacts אשר זומנו לפגישה הזו.
- numOfContacts: גודל מערך המצביעים
- location: מיקום בו תתקיים הפגישה (טקסט לא יותר מ-TEXT\_SIZE תווים)
- description: תאור הפגישה (מערך תווים שיוקצה באופן דינאמי)

### יומן (Calendar):

- owner: הבעלים של היומן, משתנה מסוג Contact
- evt: מצביע למערך של מצביעים ל-Event. אלו הם כלל הפגישות הנמצאות ביומן זה.
- evtSize: גודל מערך המצביעים

### להלן הגדרת המבנים:

```
#define TEXT_SIZE 40
typedef struct
{
    int day; //1-31
    int month; //1-12
    int year; //e.g. 2019
    int hour; //0-23
    int min; //0-59
}Date;

typedef struct
{
    char name[TEXT_SIZE];
    char phone[TEXT_SIZE];
    char emailAddress[TEXT_SIZE];
}Contact;
```



```
typedef struct
{
    char title[TEXT_SIZE];
    Date startDat;
    Date endDat;
    Contact** cont;
    int numOfContacts;
    char location[TEXT_SIZE];
    char* description;
}Event;
```

```
typedef struct
{
    Contact owner;
    Event** evt;
    int evtSize;
}Calendar;
```

### סעיף א (10 נקודות)

ממש את הפונקציה `setEvent`. הפונקציה מקבלת מצביע למערך של מצביעים ליומנים, גודל המערך ומצביע לארוע/פגישה. במידה והפגישה יכולה להתקיים (אינה בחפיפה, אפילו חלקית, עם כלל היומנים) יש לעדכן את הפגישה בכל אחד מהיומנים (מספיק להוסיף לכל יומן את המצביע לפגישה) ולהחזיר 1. במידה והפגישה אינה יכולה להתקיים, אין לשנות כלל את היומנים ולהחזיר 0.

יש לשים לב להקצאות שהנך נדרש לבצע (ניתן להניח כי כל ההקצאות הצליחו).

להלן חתימת הפונקציה:

```
int setEvent(Calendar** cal, int numCalendars, Event* ev);
```

### סעיף ב (20 נקודות)

בסעיף זה עליכם לממש את הפונקציה `findAndSetWorkMeeting`. פונקציה זו מקבלת מצביע למערך יומנים (`cal`), גודל מערך היומנים (`numCalendars`), תאריך התחלה (`date`) לתאום פגישה, משך זמן פגישה בשעות (`duration`) ומצביע לארוע/פגישה (`ev`) – השדה `startDat` ו-`endDat` ב-`ev` אינם מאותחלים וערכם שווה לאפס (כחלק מן השאלה תצטרך למצוא אותם).

על הפונקציה לאתר מועד בו הפגישה יכולה להתקיים (אינה בחפיפה, אפילו חלקית, עם כלל היומנים) בתאריך `date` החל מהשעה המצויינת בו (ב-`date`) אך הפגישה חייבת להסתיים לכל המאוחר בשעה 17:00. ניתן להניח כי השעה





ב-date עגולה – דהיינו הדקות שוות לאפס. במידה ונמצאה שעה מתאימה לכל היומנים, יש להוסיף את הפגישה המעודכנת עם טווח השעות שלה ביומנים של כלל המשתתפים ולהחזיר מצביע לפגישה ev המעודכנת.

במידה ולא נמצא מועד מתאים יש להחזיר NULL.

ניתן להניח כי הפרמטרים שהתקבלו בפונקציה חוקיים (לדוגמה:  $\text{duration} > 0$  וכו')

להלן חתימת הפונקציה:

```
Event* findAndSetWorkMeeting(Calendar** cal, int numCalendars,  
                             Date date, int duration, Event* ev);
```

**דוגמה:** מצא פגישה המתאימה לכלל היומנים, הפגישה יכולה להתחיל בתאריך 17/3/2019 בשעה 7:00, משך הפגישה שעתיים. פלט אפשרי הינו מצביע לפגישה מעודכנת (ev) בו השעה שונתה ל-11:00 (נניח שבין השעות 11:00 ל-13:00 אין פגישה לכלל בעלי היומנים שנתקבלו). לאחר שנמצאה השעה, כלל היומנים מתעדכנים בפגישה.

### סעיף ג (10 נקודות)

ממש את הפונקציה deleteContactFromEvent. הפונקציה מקבלת מצביע למערך מצביעי יומנים, מספר יומנים, מצביע (למחרוזת) לכתובת אימייל של איש קשר ומצביע לאירוע/פגישה (ev).

על הפונקציה להסיר את הפגישה ev מהיומן של איש הקשר בעל האימייל שהתקבל (תחילה עליה לאתר בעל יומן ממערך היומנים בעזרת כתובת האימייל) וכמובן לעדכן את הפגישה כך שיש פחות משתתפים בפגישה כולל המצביע אליהם.

הפונקציה תחזיר מצביע לפגישה המעודכנת.

NULL יוחזר במקרה שבפגישה שהוסרה היה איש קשר בודד, ולכן נמחקה כליל.

```
Event* deleteContactFromEvent(Calendar** cal,  
                              int numCalendars, char* email, Event* ev);
```

את הפתרון לשאלה זו יש לכתוב בקובץ Q4.c



## הוראות הגשה

Submit system v2.0

Not secure | ck.cs.colman.ac.il/test.jsp

### הגשת מבחן

cs\_intro  
moed\_a

☐ הורד פתרון  
☒ הגשה

☐ קורס  
שם המטלה  
☐ הורד מטלה  
☐ אימון

☐ הגשה סופית

אחרית לא ניתן להגיש שוב

\*\*\*\*\*  
Choose Files 4 files  
שלח

ת.ז (9 ספרות):  
קבצים:

נא להמתין בסבלנות לאחר השליחה לאישור השרת אודות העלאת הקבצים. ללא אישור מפורש של השרת ייתכן ולא נקלטו הקבצים.

הגשת המבחן תיחשב רק במידה הגשה סופית. לאחרית מתקבל מס' אסמכתא שאותו יש להגיש לביחנת ביציאה מהכיתה

נא להגיש את הקבצים \*גם\* בלינק הבא לשרת הגיבוי

[לינק לשרת הגיבוי](#)

סטטוס הגשה עבור מוד: הגשה  
פלט הקומפילר עבור כל שאלה (במידה ויש):  
אין  
פלט שגיאות ריצה (במידה ויש):  
אין  
שגיאות לוגיות:

done  
done  
done  
Question 4 will be checked manually, don't worry (-35)  
done

הציון שלך הוא: 65

את המבחן יש להגיש ל cs\_intro | moed\_a. יש למלא ת.ז עם 9 ספרות ולהוריד את המטלה.

- בכל הגשה יש להגיש את כל הקבצים Q1.c, Q2.c, Q3.c, Q4.c ורק אותם. גם אם עדין לא כתבתם את התשובות לכל השאלות עדין יש להגיש את כל הקבצים. כל שאלה נבדקת בנפרד. שימו לב לשמות הקבצים ("Q" גדולה).
- שים לב שצריך להגיש גם את Q4.c למרות שאינה נבדקת אוטומטית.
- שים לב שהציון המקסימלי הוא 60 כי שאלה 4 השווה 40 נק' תיבדק ידנית.
- רק הגשה במוד "הגשה סופית" תיחשב כהגשת המבחן.
- אחרי הגשה במוד "הגשה סופית" יתקבל מס' אסמכתא בן 4 ספרות (לא ניתן יהיה להגיש את הבחינה שוב). את מס' האסמכתא יש לשמור ולתת לבוחנת.
- לשם גיבוי הבחינה יש ללחוץ על הלינק "הגשה לשרת הגיבוי", הגישו את הפתרון גם לשם. ניתן להתעלם ממספר האסמכתא של שרת הגיבוי.

## בהצלחה