

המסלול האקדמי המכללה למינהל

ביה"ס למדעי המחשב



ת.ז. הסטודנט: _____
מספר חדר: _____
מספר נבחן: _____
מספר אסמכתא: _____

ברקוד נבחן

מבחן בקורס: מבוא למדעי המחשב

תאריך הבחינה: 15.02.18

שנת הלימודים: 2018, סמסטר: א', מועד: א'

משך הבחינה: 5 שעות

שם המתרגל/ים:

ענבל אברהם
אביב שוקרון
אורי שילה
דור ניסים
רומן סמולקין
יניב כליף

שם המרצה/ים:

ד"ר יהודה אלמליח
ד"ר גלית חיים

מבנה הבחינה: הבחינה מורכבת מחלק אחד.

מספר השאלות הכולל בבחינה: 4.

משקל כל שאלה: בצמוד לכל שאלה

הוראות לנבחן:

- אסור השימוש בכל חומר עזר
- יש לענות במחשב.
- לא נדרש להחזיר את השאלון.
- לא מצורף נספח לבחינה
- מחברת טיוטה: כן, לפי בקשת הסטודנט/ית
- מחברת נפרדת לכל שאלה: לא.

בהצלחה!!

הקדמה

במבחן זה עליכם לענות על 4/4 שאלות תכנותיות ב C. משך המבחן 5 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 קובצי ה c מושלמים. לא ב zip או דומין, אלא את קובצי 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 מיד. זהו אינו ציון המבחן ולא חלק ממנו	ציון מיד. – 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 65. 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 יהי ניתן להוריד את קובצי 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.

חומר עזר: בכוון D:\150218 תמצאו קובץ pdf המכיל ספר עזר בשפת C.

בהצלחה!

שאלה 1: 20 נקודות

כתבו את הפונקציה:

```
void calcSum(int arr_in[][M], int arr_sum[][M], int rows)
```

אשר מקבלת מערך דו מימדי `arr_in` של מספרים שלמים, ומחשבת את מערך הסכומים `arr_sum` המוגדר באופן הבא: בכל מקום במערך הסכומים יופיע הסכום של כל האיברים באותה השורה ובאותה העמודה, לא כולל האיבר עצמו.

`rows` מציין את מספר השורות במערך

לדוגמה עבור המערך `arr_in` הבא:

1 3 5 13

6 8 1 7

9 0 7 2

יש למלא את המערך `arr_sum` כך:

36 27 25 18

26 17 33 30

16 29 17 36

ממשו את הקוד הנדרש בקובץ `Q1.c`

שאלה 2: 20 נקודות

א. (10 נקודות: מחרוזות)

כתבו פונקציה המקבלת מחרוזת ומוחקת ממנה את כל הרווחים.

```
char *removeBlanks(char *str);
```

אין להקצות שטח למחרוזת חדשה אלא לעדכן את המחרוזת שהתקבלה.

לדוגמא: הפעלת הפונקציה על המחרוזת:

str: "hello Nice World"

תחזיר str: "helloNiceWorld"

יש לשים לב לעדכן '\0' בסוף המחרוזת העדכנית כמוכן יתכנו מספר רווחים ברצף.

בסעיף זה פונקציית הספרייה היחידה שניתן להשתמש הינה strlen

ב. (10 נקודות: BIT WISE)

כתבו פונקציה המקבלת מספר שלם ואינדקס (האינדקס מתחיל מימין מ 0) ומחזירה את ערך הביט במקום האנדקס.

```
int getBitByIndex(unsigned int num, int index);
```

למשל: עבור num = 6 (00000110) index = 2, התוכנית תדפיס 1, ועבור index = 3 התוכנית תדפיס 0.

ממשו את הקוד הנדרש בקובץ Q2.c

שאלה 3: 25 נקודות

סעיף א' (12 נקודות)

כתוב פונקציה רקורסיבית:

```
int toStart(int num, int start);
```

הפונקציה מקבלת מספר שלם חיובי num ומספר שלם חיובי start. הפונקציה מחזירה מספר חדש אשר הינו שרשור של המספר start עם המספר num. (יש להניח כי num ו-start גדולים מ-0).

דוגמאות:

```
toStart(45, 123) => 12345
```

```
toStart(21, 12) => 1221
```

```
toStart(12345, 6) => 612345
```

```
toStart(1,6) => 61
```

סעיף ב' (13 נקודות)

כתוב פונקציה רקורסיבית:

```
int merge(int n1, int n2);
```

הפונקציה מקבלת שני מספרים $n1, n2$ חיוביים גדולים מ-0. הפונקציה מחזירה מספר חדש המהווה מיזוג של שני המספרים מימין לסרוגין באופן הבא:

(עבור $n1$ מספר באורך 5 ו- $n2$ מספר באורך 2)

$$n1 = a_5a_4a_3a_2a_1$$
$$n2 = b_2b_1$$

יחזיר את המספר הבא:

$$a_5a_4a_3b_2a_2b_1a_1$$

דוגמאות:

```
merge(45, 123) => 12435
```

`merge(12345, 67) => 1236475`

`merge(24, 13) => 1234`

שיום/י ♥ :

השימוש בלולאות, פונקציות עזר, מערכי עזר או משתנים גלובליים/סטטים אסור, ובמידה ויהיה שימוש באחד מהנ"ל הניקוד על סעיפים אלו יהיה 0.

את הפתרון לשאלה זו יש לכתוב בקובץ Q3.c

שאלה 4(35 נק')

בשנת 2009 פתחה חברת Rovio את המשחק Angry

Birds (למעשה מדובר במשחק ה-52 אותו יצרה

החברה). Angry Birds זכה לאהדה רבה בקרב

שחקני הסמארטפון. המשחק בנוי משלבים. בכל

שלב קיימים מספר חזירים ומספר ציפורים. לכל



ציפור קיימת יכולת וכוחות שונים. החזירים מסתתרים מאחורי עצים, אבנים או קרח. על השחקן לירות ציפור בעזרת רוגטקה לעבר החזירים. בכדי לעבור שלב על השחקן להשתמש בציפורים העומדות לרשותו (בכל ציפור ניתן להשתמש פעם אחת) ולפגוע בכל החזירים הקיימים בשלב עד אשר כוחם יהיה קטן או שווה לאפס.

בשאלה זו עליך לתכנן חלק מלוגיקת משחק ה-Angry Bird.

להלן שלושה מבנים מרכזיים במשחק:

חזיר (Pig): שם (גודל לא ידוע), סטאטוס הסתתרות (1 – מאחורי עץ, 2 – מאחורי אבן, 3 – מאחורי קרח), חוסן (מספר שלם המעיד על כוחו של החזיר).

ציפור (Bird): שם (גודל לא ידוע), צבע (גודל ידוע – עד עשרה תווים), עוצמת פגיעה (מספר שלם – במידה וציפור פוגעת בחזיר, חוסנו של החזיר ירד בהתאם לעוצמת הפגיעה), יכולת חדירה (1 – רק עץ, 2 – רק אבן, 3 – רק קרח)

שלב/רמה (Level): מערך מצביעים לציפורים, גודל מערך המצביעים לציפורים, מערך מצביעים לחזירים, גודל מערך המצביעים לחזירים.

להלן הגדרת המבנים:

```
typedef struct
{
    char* name;
    int status;
    int strength;
}Pig;
```

```
typedef struct
{
    char* name;
    char color[10];
    int injuryPower;
    int penetrationType;
}Bird;
```

```
typedef struct
{
    Bird** birdsArr;
    int birdsSize;
    Pig** pigsArr;
    int pigSize;
}Level;
```

סעיף א (15 נקודות)

ממש את הפונקציה `isPassableLevel`. פונקציה זו מקבלת מצביע לשלב, על הפונקציה להחזיר 1 אם ניתן לעבור את השלב או אפס אם לא ניתן לעבור את השלב.

בכדי לדעת אם ניתן לעבור את השלב עלייך להניח כי אכן השחקן פוגע בחזירים וכל שעלייך לבדוק האם הציפורים העומדות לרשותו יספיקו לביצוע השלב.

דוגמה 1: במידה ויהיו 2 ציפורים בעלות יכולת חדירה אבן וחזיר אחד המסתתר מאחורי עץ לא תהיה יכולת לשחקן לעבור שלב זה ולכן הפונקציה תחזיר 0.

דוגמה 2: במידה ויהיה חזיר אחד עם ערך חוסן 7 (אשר מסתתר מאחורי עץ) ושני ציפורים בעלי עוצמה 3 עם יכולת חדירת עץ, לא ניתן יהיה לעבור את שלב זה – הפונקציה תחזיר 0.

דוגמה 3: במידה ויהיו 2 חזירים, האחד עם חוסן 3 ומסתתר מאחורי עץ והשני עם חוסן 3 אשר מסתתר מאחורי קרח. כמוכן יהיו 2 ציפורים בעלי עוצמה 2 (כ"א) החודרים עץ וציפור נוספת בעלת עוצמה 3 אשר חודרת קרח, ניתן יהיה לסיים לבצע את השלב הנ"ל (2 הציפורים בעלי עוצמה 2 החודרות עץ יפגעו

בחזיר אשר מסתתר מאחורי העץ והציפור החודרת קרח תפגע בחזיר
המסתתר מאחורי הקרח) – הפונקציה תחזיר 1.

להלן חתימת הפונקציה:

```
int isPassableLevel (const Level* level);
```

סעיף ב (15 נקודות)

ממש את הפונקציה `fixLevel`. הפונקציה מקבלת מצביע לשלב (`level`), ועוצמה מקסימאלית (`max`). על הפונקציה לבדוק אם ניתן לעבור את השלב (יש להשמש בפונקציה מהסעיף הקודם), במידה ולא ניתן לעבור את השלב על הפונקציה **להוסיף לשלב** מספר מינימאלי של ציפורים מתאימות כדי שניתן יהיה לעבור את השלב. אין לייצר ציפורים עם עוצמה גדולה מ-`max`. הפונקציה מחזירה מצביע לשלב המעודכן.

להלן חתימת הפונקציה:

```
Level* fixLevel(Level* level, int max);
```

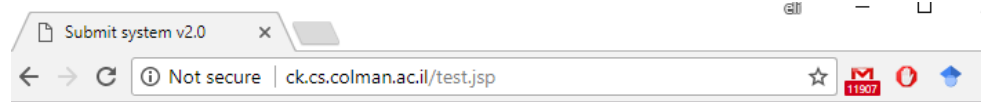
סעיף ג (5 נקודות)

ממש את הפונקציה `deleteLevel`. הפונקציה מקבלת מצביע לשלב ומוחקת את השלב וכל המידע הקשור אליו (שים לב למחיקת כל השדות שהוקצו דינאמית).

```
void deleteLevel(Level* level);
```

בהצלחה

הוראות הגשה



הגשת מבחן

cs_intro	קורס
moed_a	שם המטלה
<input type="radio"/> הורד פתרון	<input type="radio"/> הורד מטלה
<input checked="" type="radio"/> הגשה	<input type="radio"/> אימון
<input type="radio"/> הגשה סופית	
אחרית לא ניתן להגיש שוב	
*****	ת.ז (9 ספרות) :
Choose Files 4 files	קבצים :
שלח	

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

הגשת המבחן תיחשב רק במוד הגשה סופית. לאחרית מתקבל מס' אסמכתא שאותו יש להגיש לבוחנת ביציאה מהבית

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 65 כי שאלה 4 השווה 35 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