



## הפקולטה למדעי ההנדסה

המחלקה להנדסת מערכות מידע

372-1-2102 תכנות מתקדם

**תשפ"ב**

אוניברסיטת בן גוריון



**תרגיל בית מספר 3: תאריך הגשה: 21.12.21 בשעה : 23:59**

### הקדמה

בתרגיל זה ניצור מערכת המאפשרת ביצוע קרבות פוקימונים וכן [Pokedex](#) אשר יכיל מידע על הפוקימונים וסוגי פוקימונים קיימים.

תרגיל זה מחולק לשני חלקים, הראשון – בניית ה-pokedex, יתמקד בכתיבת הפוקימונים והסוגים השונים וביצוע פעולות עליהם, תוך דגש על הגדרת structs, יצירת מצביעים, הקצאות ושחרור זיכרון ועוד.

החלק השני יצירת קרבות בין פוקימונים, יתמקד בבניית ADT גנרי. תתבקשו ליצור ADT גנרי לפי הדרישות, שאינו תלוי בשימוש שייעשה בו בהמשך. בנוסף יהיה עליכם להשתמש בממשק ה-ADT כדי ליצור את מערכת הקרבות שתפורט בעבודה הבאה. אנא לפני תחילת פתירת התרגיל קראו עד הסוף את העבודה על מנת למנוע אי הבנות.

---

שלום שוב! זה אני, פרופסור C, מזמן לא דיברנו. כל הכבוד על העבודה הטובה על הפוקדקס. מכיוון שעזרתם לי מאוד, אני רוצה לנצל את היכולות שלכם פעם נוספת. ההמצאה הבאה עליה אני עובד היא מכון ללא מאמן. אני מעוניין ליצור מכוני פוקימונים ללא מאמנים על מנת לייעל את השוק. למי מכם שלא בקיא בתחום, מכון הינו מקום בו נערכים קרבות פוקימונים. המשימה שלכם היא ליצור מכון אשר יודע באופן מיידי לשלוף את הפוקימון בעל הסיכויים הכי גבוהים לנצח בכל קרב. כמו במשימה הקודמת אחוזים מהרווחים לא תקבלו, אבל ציון טוב ותהילת עולם על הפרק. בהצלחה.

---

למוטיבציה 2: [https://www.youtube.com/watch?v=Oc\\_ifA4IGIo](https://www.youtube.com/watch?v=Oc_ifA4IGIo)





## הפקולטה למדעי ההנדסה

המחלקה להנדסת מערכות מידע

372-1-2102 תכנות מתקדם

**תשפ"ב**

## אוניברסיטת בן גוריון



בחלק זה נתמקד בכתיבה נכונה של קוד, חלוקה למודלים וכתיבה של ADT גנרי. עבודה זו תתבצע בשני שלבים: בשלב הראשון עליכם ליצור ADT גנרי לפי הדרישות, שאינו תלוי בשימוש שייעשה בו בהמשך. בשלב השני, עליכם להשתמש בממשק ה-ADT שיצרתם כדי ליצור את מערכת הקרבות של הפוקימונים שלכם.

לפני שתתחילו את העבודה אנא קראו את כל ההוראות במלואן **ועבדו לפי הסדר** של העבודה על מנת שלא תאלצו לבצע עבודה כפולה ומיותרת. לנוחיותכם מצורף קובץ קונפיגורציה וכן דוגמאת הרצה מלאה של המערכת. במידה ויש סתירה כלשהי בין הכתוב בעבודה זו לבין המידע בקובץ ההרצה – יש להסתמך על הריצה המלאה שסופקה בקובץ.

### **בעבודה זו תממשו שלושה מבני נתונים גנריים שונים.**

מבני הנתונים אותם תממשו הינם רשימה מקושרת גנרית, ערמת מקסימום ומבנה נתונים של קרבות לפי סוגים. (אותו נפרט בהמשך). את שלושת מבני הנתונים יש לכתוב בצורה הטובה ביותר ונניח כי המשתמש שירצה להשתמש במודול שלנו יעשה שימוש נכון וישלח פונקציות תקינות. אין צורך לבדוק באיזשהי צורה או לדאוג לגבי מקרה שפונקציה של המשתמש שקיבלנו תגרור לשגיאה במודול שלנו.

### **:LinkedList**

- תממשו ADT של רשימה מקושרת גנרית, המוגדרת כך: על הרשימה המקושרת לתמוך בשמירת ערך עבור כל חוליה. הרשימה תתמוך בפעולות הבאות:
1. יצירה של רשימה גנרית (createLinkedList) מקבל את כל המידע אודות האובייקט הגנרי שתשמור, תאתחל רשימה ותחזיר מצביע לרשימה.
  2. הריסה של רשימה גנרית (destroyList) מקבלת מצביע לרשימה ואחראית למחיקה של הרשימה, ופינוי הזיכרון.
  3. הוספה של חוליה לרשימה (appendNode) פונקציה אשר מקבלת מצביע לרשימה ואלמנט שנרצה להוסיף, ומוסיפה אותו כחוליה לסוף הרשימה.
  4. מחיקה של חוליה מהרשימה (deleteNode) פעולה אשר מקבלת מצביע לרשימה ואלמנט ששמור כחוליה ברשימה (יש לוודא כי שמור, במידה ולא יעודכן הסטטוס של הפעולה ככישלון) ומוחקת את החוליה שבה שמור מהרשימה, ומנקה את הזיכרון שלו.
  5. הדפסת הרשימה (displayList) הפעולה מקבלת מצביע לרשימה ומדפיסה את כל האיברים ברשימה.





6. חיפוש בעזרת מפתח (searchByKeyInList) הפעולה מקבלת מצביע לרשימה ומצביע למידע מסויים (גם אלמנט) המהווה חלק מסויים מתוך האלמנט השמור, הפונקציה בודקת האם קיימת חוליה בה שמור ערך שמכיל את אותו חלק שהתקבל, במידה וכן מחזירה את הערך המלא מהרשימה, במידה ולא מחזיר NULL. אם יש יותר מאחד, יוחזר הראשון שנכנס לרשימה (רמז: על מנת לבצע את סעיף זה, יש לבקש מהמשתמש ביצירת המבנה פונקציה שתעזור לכם לבדוק את הקשר בין ערך לבין המידע שמתקבל בפונקציה. לדוגמא, אם הערך השמור הוא int, קריטריון אפשרי הוא לבדוק האם המספר מתחלק במספר אחר. במידה ויתקבל בפונקציה המספר 2 לדוגמא, פונקציה אפשרית מהמשתמש תבדוק האם ערך ברשימה מתחלק ב2 ובמידה וכן תחזיר את הראשון שתמצא)

### **MaxHeap:**

עליכם לממש ADT גנרי של ערימת מקסימום. על הערמה להתנהג על פי החוקים של ערמת מקסימום – היא תחזיק באובייקטים ותשלוף ב  $O(1)$  את האיבר המקסימלי בשעת הצורך. לערימה שם שמייחד אותה (על אף שבדרך כלל אין צורך בשם ייחודי בערימה, דרישה זו חיונית לערימה שלנו), וכל ערימה מוגבלת למספר איברים אשר יתקבל מהמשתמש ביצירת הערימה. על פי איזה קריטריון ימויינו האובייקטים יבחר המשתמש בערימה. מצורף קובץ עם פסאדו קוד ומידע נוסף מתוך קורס מבני נתונים של שנה א. על מבנה הנתונים לתמוך בפעולות הבאות:

1. יצירת ערימה (CreateHeap) פעולה אשר מקבלת את כל המידע אודות האובייקט הגנרי שיישמר. בנוסף תקבל את המספר המקסימלי של איברים בערימה, ואת השם הייחודי של הערימה. הפונקציה תיצור את המבנה נתונים כנדרש.
2. הריסה של הערימה (destroyHeap) הפעולה מקבלת מצביע לערימה. הפעולה אחראית למחיקה של הערימה, ופינוי הזיכרון של כל האיברים בערימה.
3. הכנסת איבר לערימה (insertToHeap) הפונקציה מקבלת מצביע לאלמנט ובמידה ובערימה אין איברים ככמות המכסה האפשרית, תכניס את האיבר לערימה. יש להכניס את האיבר על פי חוקי הערימה.
4. הדפסת הערימה (printHeap) מקבלת מצביע לערימה ומדפיסה את כל האברים על פי סדרם מהגדול לקטן. על ההדפסה להיות בפורמט הבא:

\*name\*:

1. \*elementDetails\* 2. \*elementDetails\*

- שימו לב פרטי ההדפסה של האיברים הוא על פי החלטתו של המשתמש.





## הפקולטה למדעי ההנדסה

המחלקה להנדסת מערכות מידע  
372-1-2102 תכנות מתקדם  
תשפ"ב

## אוניברסיטת בן גוריון



כלומר: שם הערימה, לאחריו נקודותיים וירידת שורה. לאחר מכן יודפס מספר סידורי נקודה ולאחריו רווח. לאחר הרווח יודפס הפרטים של האלמנט. וכך הלאה. (אין פה ירידת שורה כלל. המשתמש מחליט איך יודפס עבורו המידע, ולכן תהיה ירידת שורה, או אפילו שתיים, רק במקרה שהמשתמש בחר שבהדפסת האלמנט שלו יהיה ירידה של שורה)

במידה ואין איברים בערימה, יודפס השורות הבאות:

\*name\*:

No elements.

יודפס שם הערימה, לאחריו נקודותיים וירידת שורה. לאחר הדפסת השורה של הערימה הריקה יבואו שתי ירידות שורה (כלומר ח\n\n).

"No elements.\n\n"

5. החזרת ערך מקסימלי והוצאתו (PopMaxHeap) הפעולה מקבלת מצביע לערימה, הפונקציה תחזיר את האיבר המקסימלי בערימה ותמחק אותו מהערימה. במידה והערימה ריקה, יוחזר NULL.
  6. החזרת ערך מקסימלי (TopMaxHeap) הפעולה מקבלת מצביע לערימה, הפונקציה תחזיר מצביע לאיבר המקסימלי בערימה. במידה והערימה ריקה, יוחזר NULL.
  7. החזרת שם הערימה (getHeapId).
  8. החזרת מספר האיברים הקיימים כרגע בתוך הערימה. (getHeapCurrentSize)
- שימו לב – יש צורך ליצור פונקציות נוספות פנימיות של heapify על מנת לשמור על המבנה הנכון של מבנה הנתונים.

### :BattleByCategory

אתם מקבלים בעבודה זו את קובץ ה.H של אובייקט זה ותצטרכו לממש את הפונקציות לפי החתימות שתקבלו. המלצה – עברו על ההוראות ביחד עם החתימות של הפונקציות. שימו לב – אין לשנות דבר (!) בקובץ זה. אתם כלל לא מגישים קובץ זה, ולכן במידה ותבצעו בו שינויים הקוד שלכם לא יתקמפל. השקיעו מחשבה ב design נכון עבור מבנה נתונים זה שיהיה יעיל ככל הניתן וישתמש באופן נכון במבני הנתונים האחרים שבניתם. עליכם לממש ADT של קרבות לפי קטגוריות, המבנה יהיה גנרי ויוגדר באופן הבא: על המבנה נתונים לתמוך בשמירה של אובייקטים על פי קטגוריות.





## הפקולטה למדעי ההנדסה

המחלקה להנדסת מערכות מידע  
372-1-2102 תכנות מתקדם  
תשפ"ב

## אוניברסיטת בן גוריון



כל אובייקט שרוצה להשתמש במבנה נתונים זה חייב להכיל קטגוריה שתתואר על ידי שרשרת תווים, ומתקפה לקרב שיאופיין על ידי מספר שלם. למבנה יש מכסה לכמות האובייקטים הנשמרים מכל קטגוריה. כלומר ביצירה של קרבות יתקבל מספר (לדוגמא 20) מספר זה אומר שבכל קטגוריה יכולים להיות עד 20 אובייקטים. בנוסף כאשר יוצרים קרבות המשתמש יכניס את מספר הקטגוריות שהאובייקטים שלו מחולקים על פיהם ובנוסף שורה של שמות הקטגוריות מופרדות על ידי פסיקים. כלומר "cat1,cat2,cat3".

מבנה הנתונים מאפשר לשמור את האובייקטים לפי קטגוריות, מאפשר הדפסה למסך של כל האובייקטים, הוספה של אובייקט חדש מסוג מסויים ומחיקה של האובייקט החזק ביותר מסוג מסויים. בנוסף בהינתן אובייקט חדש, מערכת הקרבות תאפשר לבצע קרב בין אותו אובייקט לאובייקט שיש לו הסיכוי הגדול ביותר לנצח בקרב מבין האובייקטים השמורים במערכת. לוגיקת הקרב תפורט בהמשך.

בADT זה ניתן להשתמש במערך רק של משתנים פרימיטיביים (int, char, double) או מערך של פוינטרים למשתנים פרימיטיביים (int\*,char\*,double\*). עבור שאר המידע שלכם יש להשתמש במבני הנתונים הגנריים שכתבתם קודם.

**שימו לב: אתם חייבים להשתמש במבני הנתונים שכבר כתבתם – הרשימה המקושרת וכן ערימת מקסימום על מנת לממש את מבנה זה. אין להשתמש במידע "פנימי" של האובייקטים אלא רק במידע החשוף בעזרת קובץ header.**

BattleByCategory הגנרי יתמוך בפעולות הבאות:

1. יצירה של קרבות לפי קטגוריות (createBattleByCategory) פעולה אשר מקבלת את כל המידע אודות האובייקטים שישמרו במערכת. בנוסף, הפעולה תקבל את מספר הקטגוריות וכן את שמות הקטגוריות בשורה מופרדת על ידי פסיקים. נוסף על כך הפונקציה תקבל את מספר הקיבולת של האובייקטים מכל קטגוריה. הפונקציה תיצור את האובייקט ותחזיר מצביע אליו.
2. הריסה של קרבות לפי קטגוריות (destroyBattleByCategory) פעולה תקבל מצביע למבנה נתונים ותהיה אחראית למחיקה של המבנה, כל האובייקטים השמורים בו ופינוי הזיכרון.
3. הוספה של אובייקט למערכת – (insertObject) הפעולה מקבלת מצביע למבנה הנתונים ומצביע לאובייקט אותו יש להוסיף למערכת. הפעולה תקבל אובייקט בעל קטגוריה. במידה והקטגוריה מופיעה כבר באובייקט הקרבות, האובייקט יתווסף למערכת. במידה והקטגוריה לא קיימת, הוא לא יתווסף ויתקבל סטטוס כישלון.







## הפקולטה למדעי ההנדסה

המחלקה להנדסת מערכות מידע

372-1-2102 תכנות מתקדם

**תשפ"ב**

## אוניברסיטת בן גוריון



במידה ומספר האובייקטים עבור אותה קטגוריה הגיע למכסה האובייקט לא יתווסף גם כן, ויתקבל סטטוס כישלון.

4. הדפסה של כל האיברים באובייקט הקרבות (`displayObjectsByCategories`)

פעולה המקבלת מצביע לאובייקט הקרבות ומדפיסה את כל האובייקטים ששמורים לפי קטגוריות. האובייקטים יודפסו באופן מסודר מהחזק ביותר לחלש ביותר. כלומר, בהינתן 3 קטגוריות `cat1, cat2, cat3` כאשר לקטגוריה הראשונה 2 אובייקטים, לשניה אובייקט אחד ולשלישי אין אובייקטים ההדפסה תראה בצורה הבאה:

`cat1:`

1. `*elementDetails*2. *elementDetails*cat2:`

1. `*elementDetails*cat3:`

No elements.

5. מחיקה של האובייקט החזק ביותר בקטגוריה מסויימת (`removeMaxByCategory`)

פעולה המקבלת מצביע למבנה הקרבות ושם של קטגוריה. הפעולה מוחקת את האובייקט החזק ביותר מאותה קטגוריה (במידה וקיים) ומחזירה אותו למשתמש. במידה ולא קיים, יוחזר NULL.

6. החזרה של מספר האובייקטים בקטגוריה מסויימת

(`getNumberOfObjectsInCategory`) פעולה המקבלת מצביע למבנה הקרבות ושם של קטגוריה. הפעולה מחזירה כמה אובייקטים שמורים במבנה מאותה קטגוריה שהתקבלה.

7. ביצוע קרב (`fight`) הפעולה מקבלת מצביע למבנה הקרב, ואובייקט המתמודד לקרב

נגד המערכת. ביצוע הקרב יהיה בצורה הבאה: המערכת תמצא את האובייקט המתאים ביותר לקרב וזה שינצח את הקרב בצורה הטובה ביותר. בקרב יש להתחשב בקטגוריות של כל אחד מהאובייקטים, זאת מכיוון שלפי הקטגוריה עשויה להשתנות המתקפה. כלומר יש לתשאל אובייקט מה המתקפה שלו בהינתן האובייקט השני. לאחר מציאת האובייקט בעל המתקפה החזקה ביותר בהינתן האובייקט המוצע לקרב, יש לבדוק לאיזה מהאובייקטים מבין השניים המתקפה החזקה ביותר. אותו אובייקט הוא המנצח בקרב. במידה והמתקפה זהה בין האובייקט הנבחר לאובייקט המתקבל לקרב, יהיה תיקו. במידה ולקרב יש צורך לבצע שובר שיוון בין שני פוקימונים או יותר בעלי מתקפה זהה לקרב מסויים יבחר לקרב הפוקימון מהקטגוריה שכתובה קודם ברשימת





## הפקולטה למדעי ההנדסה

המחלקה להנדסת מערכות מידע

372-1-2102 תכנות מתקדם

### תשפ"ב

## אוניברסיטת בן גוריון



הקטגוריות שמתקבלת ביצירה של האובייקט (כלומר categories שמתקבל

ב createBattleByCategory).

אם לא קיימים אובייקטים שניתן להלחם נגדם יוחזר NULL. אחרת יוחזר המנצח.

במידה והיה תיקו, יוחזר האובייקט שנבחר מתוך המערכת.

במידה ויש כמה אובייקטים מקטגוריות שונות בעלות אותה מתקפה מירבית לקרב,

יבחר האובייקט מהקטגוריה שנכנסה ראשונה ברשימת קטגוריות ביצירת הADT.

ניתן להניח כי לא יהיו שני אובייקטים באותה קטגוריה בעלי אותה מתקפה.

על הפונקציה להדפיס את פרטי הקרב.

ההדפסה תראה בצורה הבאה:

The final battle between:

\*the given objects details\*

In this battle his attack is : \*attack1\*

against \*the chosen objects details\*

In this battle his attack is : \*attack2\*

THE WINNER IS:

\*winning object details\*

פירוט הדפסות:

"The final battle between:\n"

"In this battle his attack is :%d\n\n"

"against "

"In this battle his attack is :%d\n\n"

"THE WINNER IS:\n"

במקרה של תיקו, ההדפסה תראה בצורה הבאה:

The final battle between:

\*the given objects details\*

In this battle his attack is : \*attack1\*

against \*the chosen objects details\*

In this battle his attack is : \*attack2\*





הפקולטה למדעי ההנדסה

המחלקה להנדסת מערכות מידע

372-1-2102 תכנות מתקדם

**תשפ"ב**

אוניברסיטת בן גוריון



IT IS A DRAW.

פירוט הדפסות:

"The final battle between:\n"

"In this battle his attack is :%d\n\n"

"against "

"In this battle his attack is :%d\n\n"

"IT IS A DRAW.\n"

• **מבנה נתונים זה צריך לעמוד באופן עצמאי ללא קשר לפוקימונים. כל**

משתמש יכול לבחור איזה אובייקטים יהיו בו. אתם כמשתמשים תשתמשו בו בהמשך כדי לבצע את הקרבות עבור אובייקטים של פוקימונים.

**תפריט ראשי: (שימו לב כי הmain הוא ה"משתמש" שלנו בכל המודולים ומבני הנתונים**

**שיצרנו – יש לעשות הפרדה בין כל מודול כאשר כותבים את הקוד)**

**הרצה של התוכנית:**

עליכם לבנות קובץ MainSecond.c שישתמש בכל הפונקציות בתוארו לעיל. הפונקציה

תקרא בעזרת הפקודה:

`./PokemonsBattles <numberOfTypes> <capacityInType> <configurationFile>`

כאשר

- `numberOfTypes` הינו מספר שלם אשר מתאר את כמות סוגי הפוקימונים איתם נעבוד, ואותם יש לקרוא מן הקובץ (לדוגמא 3)
  - `capacityInType` הינו מספר שלם אשר מתאר את כמות הפוקימונים המקסימלי שניתן להכניס למערכת הקרבות בכל אחד מהסוגים השונים (לדוגמא 20)
  - `configurationFile` הינו נתיב לקובץ קונפיגורציה של סוגי פוקימונים ופוקימונים אותם יש לקרוא מתוך הקובץ ולהכניסם למערכת. (אותו דבר מהעבודה הקודמת)
- \*שימו לב כי הפעם – לא מתקבל מספר הפוקימונים שמופיעים בקובץ הקונפיגורציה!**







## הפקולטה למדעי ההנדסה

המחלקה להנדסת מערכות מידע

372-1-2102 תכנות מתקדם

תשפ"ב

אוניברסיטת בן גוריון



קובץ הקונפיגורציה מוגדר בצורה הבאה:

Types

Type1,Type2,Type3

Type1 effective-against-me: Type2,Type3

Type1 effective-against-other: Type3

Type2 effective-against-other: Type1

Pokemons

pokemonName1,PokemonSpecies1,PokemonHeight1,PokemonWeight1,PokemonAttack1,PokemonType1

pokemonName2,PokemonSpecies2,PokemonHeight2,PokemonWeight2,PokemonAttack2,PokemonType2

שימו לב – את תחילת הסוגים יש לזהות לפי הכותרת Types. אחריה יופיעו רשימה של הסוגים על פי הכמות שניתנה בשורת הרצה. בין השורות יפרידו פסיקים. לאחר מכן יופיע המידע שיש להוסיף לגבי הקשרים בין הסוגים בקרב. אין חשיבות לסדר בו יופיעו ואין בהכרח קשר סימטרי – כלומר אם כתוב אצל Fire ש Water אפקטיבי נגדו בקרב, לא בהכרח יהיה כתוב אצל Water שהוא אפקטיבי בקרב נגד Fire. אין מניעה כי המידע יופיע בשתי שורות נפרדות. כלומר:

Type1 effective-against-me:Type2

Type1 effective-against-me:Type3

הסוג שמופיע בתחילת השורה הינו סוג המקור, במידה והשורה מסוג **effective-against-me**, הסוגים ברשימה הינם אפקטיבים נגד סוג המקור בקרב. במידה והשורה מסוג **effective-against-other** המקור אפקטיבי נגד סוגים אלו בקרב.

לאחר המידע הנ"ל שלא בהכרח יופיע אודות כל הסוגים, יופיע הכותרת Pokemons ומתחתיה כל הפוקימונים שיש להכניס למערכת על פי הפורמט שנכתב. אין רווחים כלל בין הפרטים והם מופרדים בפסיקים. ניתן להניח את תקינות המידע בקובץ הקונפיגורציה.





הפקולטה למדעי ההנדסה

המחלקה להנדסת מערכות מידע

372-1-2102 תכנות מתקדם

**תשפ"ב**

אוניברסיטת בן גוריון



להלן דוגמא אפשרית לקובץ קונפיגורציה כזה:

Types

Fire,Water,Electric,Grass

Fire effective-against-me:Water

Fire effective-against-other:Grass,Electric

Electric effective-against-other:Water

Pokemons

Pikachu,MousePokemon,0.4,6,55,Electric

Charmander,LizardPokemon,0.6,8.5,52,Fire

Ponyta,FireHorsePokemon,1.0,30.0,85,Fire

Squirtle,TinyTurtlePokemon,0.5,9,48,Water

Psyduck,DuckPokemon,0.8,19.6,52,Water

שימו לב – שם של פוקימון או סוג תמיד יהיה מילה אחת ללא רווחים. כנל לגבי זן הפוקימון.

קובץ הקונפיגורציה שינתן לכם יהיה **תקין לוגית**. כלומר, לא ינתנו פוקימונים עם סוג שלא

הוגדר. לא יוגדרו פוקימונים מסוג מסויים מעל המכסה שמותר להחזיק. לא יהיה סוג של

פוקימון פעמים וכו. בנוסף, ניתן להניח כי לא יהיה בקובץ קונפיגורציה כמות של פוקימונים

מסוג מסויים מעל המכסה המותרת לסוג.

ייתכן קובץ ריק ממידע. במקרה זה לא יהיו פוקימונים ולא סוגים של פוקימונים יהיו רק

הכותרות :

Types

Pokemons





הפקולטה למדעי ההנדסה

המחלקה להנדסת מערכות מידע

372-1-2102 תכנות מתקדם

תשפ"ב

אוניברסיטת בן גוריון



### Main - פונקציונליות:

כאשר התוכנית מתחילה היא תחילה קוראת מן הקובץ את כל המידע שפורט – במידע בקובץ הוא מסד הנתונים שלנו. שימו לב שאת המידע יש לאחסן בעזרת אובייקט ה BattleByCategory. לאחר מכן יוצג למשתמש הודעת פתיחה הבאה:

Please choose one of the following numbers:

- 1 : Print all Pokemons by types
- 2 : Print all Pokemons types
- 3 : Insert Pokemon to battles training camp
- 4 : Remove strongest Pokemon by type
- 5 : Fight
- 6 : Exit

בסוף כל שורה יש ירידת שורה אחת בדיוק ח).

התוכנית תחכה לקלט מהמשתמש ולאחריו אנטר. הקלדת תו שונה מהאפשרויות הנל תגרור הודעת שגיאה למסך, והתפריט יוצג שוב. דוגמא:

Please choose one of the following numbers:

- 1 : Print all Pokemons by types
- 2 : Print all Pokemons types
- 3 : Insert Pokemon to battles training camp
- 4 : Remove strongest Pokemon by type
- 5 : Fight
- 6 : Exit

i

Please choose a valid number.

Please choose one of the following numbers:

- 1 : Print all Pokemons by types
- 2 : Print all Pokemons types





## הפקולטה למדעי ההנדסה

המחלקה להנדסת מערכות מידע

372-1-2102 תכנות מתקדם

**תשפ"ב**

אוניברסיטת בן גוריון



3 : Insert Pokemon to battles training camp

4 : Remove strongest Pokemon by type

5 : Fight

6 : Exit

פירוט הדפסות:

"Please choose a valid number.\n"

פונקציונליות הפקודות:

1) Print all Pokemons by types – מדפיס את כל המידע אודות הפוקימונים שנמצאים במערכת הקרבות. לאחר מכן יוצג שוב התפריט. סדר ההדפסה יהיה לפי קטגוריות ועל פי הסדר של הקטגוריות בקובץ הקונפיגורציה. הפוקימונים יודפסו בתוך הקטגוריות על פי חוזק המתקפה שלהם. הראשון יהיה הפוקימון החזק ביותר הבא אחריו השני החזק ביותר וכך הלאה. פורמט ההדפסה כפי שמופרט לעיל בפונקציות הדפסה.

דוגמא:

Please choose one of the following numbers:

1 : Print all Pokemons by types

2 : Print all Pokemons types

3 : Insert Pokemon to battles training camp

4 : Remove strongest Pokemon by type

5 : Fight

6 : Exit

**1**

Fire:

1 .Ponyta:

FireHorsePokemon, Fire Type.

Height: 1.00 m   Weight: 30.00 kg   Attack: 85

2 .Growlithe:

PuppyPokemon, Fire Type.

Height: 0.70 m   Weight: 19.00 kg   Attack: 70





## הפקולטה למדעי ההנדסה

המחלקה להנדסת מערכות מידע

372-1-2102 תכנות מתקדם

**תשפ"ב**

אוניברסיטת בן גוריון



3. Charmander:

LizardPokemon, Fire Type.

Height: 0.60 m    Weight: 8.50 kg    Attack: 52

Water:

1. Psyduck:

DuckPokemon, Water Type.

Height: 0.80 m    Weight: 19.60 kg    Attack: 52

2. Squirtle:

TinyTurtlePokemon, Water Type.

Height: 0.50 m    Weight: 9.00 kg    Attack: 48

Electric:

1. Pikachu:

MousePokemon, Electric Type.

Height: 0.40 m    Weight: 6.00 kg    Attack: 55

Grass:

No elements.

Print all Pokemons types (2) - מדפיס את כל המידע אודות סוגי הפוקימונים שנמצאים במערכת הקרבות. לאחר מכן יוצג שוב התפריט. סדר ההדפסה יהיה על פי סדר השורות של הסוגים בקובץ הקונפיגורציה. שימו לב – יש לתחזק את מספר הפוקימונים בכל אחת מהקטגוריות. אם במהלך העבודה נוספים פוקימונים או נמחקים פוקימונים מספר הפוקימונים מכל קטגוריה צריך להשתנות בהתאם.

פורמט ההדפסה כפי שמופרט בפונקציית הדפסה של סוגי פוקימון.





## הפקולטה למדעי ההנדסה

המחלקה להנדסת מערכות מידע

372-1-2102 תכנות מתקדם

**תשפ"ב**

אוניברסיטת בן גוריון



דוגמא:

Please choose one of the following numbers:

- 1 : Print all Pokemons by types
- 2 : Print all Pokemons types
- 3 : Insert Pokemon to battles training camp
- 4 : Remove strongest Pokemon by type
- 5 : Fight
- 6 : Exit

**2**

Type Fire -- 3 pokemons

These types are super-effective against Fire:Water ,Electric

Fire moves are super-effective against:Grass

Type Water -- 2 pokemons

Type Electric -- 1 pokemons

Electric moves are super-effective against:Fire

Type Grass -- 0 pokemons

(3 Insert Pokemon to battles training camp - פונקציונליות זו מאפשרת להכניס פוקימון נוסף אל הפוקימונים השמורים במערכת הקרבות. ניתן להניח תקינות קלט של המידע אודות הפוקימון שנכנס! . לאחר בחירה באפשרות זו המערכת תדפיס למסך הודעה המבקשת לדעת איזה סוג של פוקימון תרצו להכניס למערכת. במידה והסוג אינו קיים תוצג הודעת שגיאה והתפריט יודפס בשנית. במידה והוא קיים המערכת תבקש להכניס את כל המידע שהכרחי ליצירת פוקימון. לאחר מכן המערכת תכניס את הפוקימון למערכת הקרבות ותציג הודעה שהפוקימון נוסף בהצלחה. שימו לב – אין בעיה שפוקימון יופיע יותר מפעם אחת עם נתונים שונים. במידה ונוצר פוקימון חדש בסוף ההכנסה יודפס למסך המידע על הפוקימון הנוצר.

לדוגמא:

Please choose one of the following numbers:

- 1 : Print all Pokemons by types







הפקולטה למדעי ההנדסה

המחלקה להנדסת מערכות מידע

372-1-2102 תכנות מתקדם

תשפ"ב

אוניברסיטת בן גוריון



2 : Print all Pokemons types

3 : Insert Pokemon to battles training camp

4 : Remove strongest Pokemon by type

5 : Fight

6 : Exit

3

Please enter Pokemon type name:

notExist

Type name doesn't exist.

Please choose one of the following numbers:

1 : Print all Pokemons by types

2 : Print all Pokemons types

3 : Insert Pokemon to battles training camp

4 : Remove strongest Pokemon by type

5 : Fight

6 : Exit

3

Please enter Pokemon type name:

Grass

Please enter Pokemon name:

Oddish

Please enter Pokemon species:

WeedPokemon

Please enter Pokemon height:

0.5

Please enter Pokemon weight:

5.4

Please enter Pokemon attack:

50

The Pokemon was successfully added.

Oddish :

WeedPokemon, Grass Type.





הפקולטה למדעי ההנדסה

המחלקה להנדסת מערכות מידע

372-1-2102 תכנות מתקדם

**תשפ"ב**

אוניברסיטת בן גוריון



Height: 0.50 m   Weight: 5.40 kg   Attack: 50

פירוט ההדפסות האפשריות:

"Please enter Pokemon type name:\n"

"Type name doesn't exist.\n"

"Please enter Pokemon name:\n"

"Please enter Pokemon species:\n"

"Please enter Pokemon height:\n"

"Please enter Pokemon weight:\n"

"Please enter Pokemon attack:\n"

"The Pokemon was successfully added.\n"

"Type at full capacity.\n"

4) Remove strongest Pokemon by type – לאחר בחירה באפשרות זו המערכת

תדפיס הודעה המבקשת להזין סוג של פוקימון ממנו רוצים למחוק את הפוקימון

החזק ביותר מהמערכת. במידה והסוג אינו קיים תוצג הודעת שגיאה. במידה והסוג

קיים במערכת אך אין פוקימונים מסוג זה, תוצג הודעת שגיאה. במידה וישנם

פוקימונים מסוג זה, המערכת תוציא ממערכת הקרבות את הפוקימון החזק ביותר

ותדפיס למסך מי נמחק.

לדוגמא:

Please choose one of the following numbers:

1 : Print all Pokemons by types

2 : Print all Pokemons types

3 : Insert Pokemon to battles training camp

4 : Remove strongest Pokemon by type

5 : Fight

6 : Exit

4

Please enter type name:

notExist

Type name doesn't exist.

Please choose one of the following numbers:





## הפקולטה למדעי ההנדסה

המחלקה להנדסת מערכות מידע

372-1-2102 תכנות מתקדם

**תשפ"ב**

אוניברסיטת בן גוריון



- 1 : Print all Pokemons by types
- 2 : Print all Pokemons types
- 3 : Insert Pokemon to battles training camp
- 4 : Remove strongest Pokemon by type
- 5 : Fight
- 6 : Exit

**4**

Please enter type name:

**Fire**

The strongest Pokemon was removed:

Ponyta :

FireHorsePokemon, Fire Type.

Height: 1.00 m   Weight: 30.00 kg   Attack: 85

פירוט ההדפסות האפשריות:

"Please enter type name:\n"

"Type name doesn't exist.\n"

"The strongest Pokemon was removed:\n"

"There is no Pokemon to remove.\n"

(5) Fight - לאחר בחירה באפשרות זאת, עליכם לאפשר יצירת פוקימון איתו תרצו להלחם נגד אחד הפוקימונים השמורים במערכת הקרבות. ניתן להניח תקינות קלט של המידע אודות הפוקימון שנכנס! . לאחר בחירה באפשרות זו המערכת תדפיס למסך הודעה המבקשת לדעת איזה סוג של פוקימון תרצו להכניס למערכת. במידה והסוג אינו קיים תוצג הודעת שגיאה והתפריט יודפס בשנית. במידה והוא קיים המערכת תבקש להכניס את כל המידע שהכרחי ליצירת פוקימון.

**שימו לב** – הפוקימון שנוצר לא מתווסף לפוקימונים הקיימים, הוא יהיה במערכת רק לקרב ולאחר מכן יש לפנות אותו מהזיכרון.

לאחר יצירת הפוקימון המערכת תבחר את הפוקימון שנגדו יערך הקרב מבין הפוקימונים השמורים במערכת, תנהל מולו קרב ותדפיס מידע אודות הקרב. במידה ואין פוקימונים במערכת הקרבות תוצג הודעת שגיאה שלא ניתן לבצע את הקרב.

- שימו לב: לסוגים של הפוקימונים יש משמעות בשלב הקרבות. המתקפה של כל פוקימון עלולה להשתנות בהתאם לפוקימון נגדו הוא נלחם. להזכירכם, לכל סוג





## הפקולטה למדעי ההנדסה

המחלקה להנדסת מערכות מידע  
372-1-2102 תכנות מתקדם  
תשפ"ב

## אוניברסיטת בן גוריון



פוקימון יש רשימה של סוגים אשר אפקטיביים נגדו בקרב  
(effective\_against\_me), ורשימה של סוגים אשר סוג המקור אפקטיבי נגדם  
(effective\_against\_others). בנוסף אין סימטריה בהכרח בין הסוגים.  
כלומר אם ברשימה של סוג Fire כתוב כי Water אפקטיבי נגדו בקרב, לא  
בהכרח שברשימה של Water נגד מי הוא אפקטיבי בקרב ימצא Fire.  
יש לחשב את המתקפה של כל פוקימון לפי האם הפוקימון שמולו אפקטיבי נגדו  
בקרב. במידה וכן, יפחתו 10 נקודות מהמתקפה שלו. כלומר, במידה ויש פוקימון  
מסוג Fire עם מתקפה של 50, ופוקימון של Water עם מתקפה של 40, וידוע כי  
Water אפקטיבי נגד Fire בקרב, יפחת מהפוקימון מסוג Fire 10 נקודות  
מהמתקפה. כלומר במקום 50 המתקפה שלו תהיה 40. במקרה זה, לשני  
הפוקימונים תהיה מתקפה של 40 ולכן יהיה תיקו בקרב.  
לא ירד לפוקימון יותר מ-10 נקודות, במידה וברשימה של סוג א של פוקימונים  
שאפקטיביים נגדו (נגד סוג א) יופיע סוג ב או שברשימה של סוג ב ברשימה של  
הסוגים שהוא (סוג ב) אפקטיבי נגדם יופיע סוג א ירד לפוקימון מסוג א 10  
נקודות. גם אם הוא מופיע בשתי הרשימות הללו ירדו רק 10 נקודות.  
לדוגמא:  
עבור המידע הנל לגבי הסוגים מקובץ הקונפיגורציה:

### Types

Fire,Water,Electric,Grass

Fire effective-against-me:Water,Electric

Fire effective-against-other:Grass

Electric effective-against-other:Fire

Please choose one of the following numbers:

- 1 : Print all Pokemons by types
- 2 : Print all Pokemons types
- 3 : Insert Pokemon to battles training camp
- 4 : Remove strongest Pokemon by type
- 5 : Fight
- 6 : Exit

5

Please enter Pokemon type name:





הפקולטה למדעי ההנדסה

המחלקה להנדסת מערכות מידע

372-1-2102 תכנות מתקדם

תשפ"ב

אוניברסיטת בן גוריון



notExist

Type name doesn't exist.

Please choose one of the following numbers:

1 : Print all Pokemons by types

2 : Print all Pokemons types

3 : Insert Pokemon to battles training camp

4 : Remove strongest Pokemon by type

5 : Fight

6 : Exit

5

Please enter Pokemon type name:

Fire

Please enter Pokemon name:

Charizard

Please enter Pokemon species:

FlamePokemon

Please enter Pokemon height:

1.7

Please enter Pokemon weight:

90.5

Please enter Pokemon attack:

90

You choose to fight with:

Charizard :

FlamePokemon, Fire Type.

Height: 1.70 m   Weight: 90.50 kg   Attack: 90

The final battle between:

Charizard :

FlamePokemon, Fire Type.

Height: 1.70 m   Weight: 90.50 kg   Attack: 90





הפקולטה למדעי ההנדסה

המחלקה להנדסת מערכות מידע

372-1-2102 תכנות מתקדם

תשפ"ב

אוניברסיטת בן גוריון



In this battle his attack is :90

against Magmar :

SpitfirePokemon, Fire Type.

Height: 1.30 m Weight: 44.50 kg Attack: 80

In this battle his attack is :80

THE WINNER IS:

Charizard :

FlamePokemon, Fire Type.

Height: 1.70 m Weight: 90.50 kg Attack: 90

Please choose one of the following numbers:

1 : Print all Pokemons by types

2 : Print all Pokemons types

3 : Insert Pokemon to battles training camp

4 : Remove strongest Pokemon by type

5 : Fight

6 : Exit

5

Please enter Pokemon type name:

Fire

Please enter Pokemon name:

Vulpix

Please enter Pokemon species:

FoxPokemon

Please enter Pokemon height:

0.6

Please enter Pokemon weight:

9.9

Please enter Pokemon attack:







הפקולטה למדעי ההנדסה

המחלקה להנדסת מערכות מידע

372-1-2102 תכנות מתקדם

תשפ"ב

אוניברסיטת בן גוריון



42

You choose to fight with:

Vulpix :

FoxPokemon, Fire Type.

Height: 0.60 m   Weight: 9.90 kg   Attack: 42

The final battle between:

Vulpix :

FoxPokemon, Fire Type.

Height: 0.60 m   Weight: 9.90 kg   Attack: 42

In this battle his attack is :42

against Magmar :

SpitfirePokemon, Fire Type.

Height: 1.30 m   Weight: 44.50 kg   Attack: 80

In this battle his attack is :80

THE WINNER IS:

Magmar :

SpitfirePokemon, Fire Type.

Height: 1.30 m   Weight: 44.50 kg   Attack: 80

Please choose one of the following numbers:

1 : Print all Pokemons by types

2 : Print all Pokemons types

3 : Insert Pokemon to battles training camp

4 : Remove strongest Pokemon by type

5 : Fight

6 : Exit

5





הפקולטה למדעי ההנדסה

המחלקה להנדסת מערכות מידע

372-1-2102 תכנות מתקדם

תשפ"ב

אוניברסיטת בן גוריון



Please enter Pokemon type name:

Water

Please enter Pokemon name:

Lapras

Please enter Pokemon species:

TransportPokemon

Please enter Pokemon height:

2.5

Please enter Pokemon weight:

220

Please enter Pokemon attack:

70

You choose to fight with:

Lapras :

TransportPokemon, Water Type.

Height: 2.50 m Weight: 220.00 kg Attack: 70

The final battle between:

Lapras :

TransportPokemon, Water Type.

Height: 2.50 m Weight: 220.00 kg Attack: 70

In this battle his attack is :70

against Magmar :

SpitfirePokemon, Fire Type.

Height: 1.30 m Weight: 44.50 kg Attack: 80

In this battle his attack is :70

IT IS A DRAW.





הפקולטה למדעי ההנדסה

המחלקה להנדסת מערכות מידע

372-1-2102 תכנות מתקדם

תשפ"ב

אוניברסיטת בן גוריון



במקרה שאין פוקימונים במערכת:

Please choose one of the following numbers:

1 : Print all Pokemons by types

2 : Print all Pokemons types

3 : Insert Pokemon to battles training camp

4 : Remove strongest Pokemon by type

5 : Fight

6 : Exit

5

Please enter Pokemon type name:

Electric

Please enter Pokemon name:

Magneton

Please enter Pokemon species:

MagnetPokemon

Please enter Pokemon height:

1

Please enter Pokemon weight:

60

Please enter Pokemon attack:

50

You choose to fight with:

Magneton:

MagnetPokemon, Electric Type.

Height: 1.00 m   Weight: 60.00 kg   Attack: 50

There are no Pokemons to fight against you.





הפקולטה למדעי ההנדסה

המחלקה להנדסת מערכות מידע

372-1-2102 תכנות מתקדם

**תשפ"ב**

אוניברסיטת בן גוריון



פירוט ההדפסות האפשריות:

"Please enter Pokemon type name:\n"  
"Type name doesn't exist.\n"  
"Please enter Pokemon name:\n"  
"Please enter Pokemon species:\n"  
"Please enter Pokemon height:\n"  
"Please enter Pokemon weight:\n"  
"Please enter Pokemon attack:\n"  
"You choose to fight with:\n"  
"There are no Pokemons to fight against you.\n"

Exit (6) - יציאה מסודרת מהמערכת – כולל מחיקה וניקוי הזיכרון. לאחר ביצוע הניקיון תוצג הודעה למשתמש כי הכל נמחק והמערכת מבצעת יציאה.  
לדוגמא:

Please choose one of the following numbers:

- 1 : Print all Pokemons by types
- 2 : Print all Pokemons types
- 3 : Insert Pokemon to battles training camp
- 4 : Remove strongest Pokemon by type
- 5 : Fight
- 6 : Exit

6

All the memory cleaned and the program is safely closed.

פירוט ההדפסה:

"All the memory cleaned and the program is safely closed.\n"

- במקרה שבמהלך התוכנית בניסיון לשימוש בהקצאת זכרון דינאמי, היה כשלון, יוצג למשתמש ההודעה הבאה:

"No memory available.\n"

לאחר מכן יש לנקות את הזכרון שהוקצה במערכת בצורה הטובה ביותר ולצאת

מהתוכנית.





## הפקולטה למדעי ההנדסה

המחלקה להנדסת מערכות מידע  
372-1-2102 תכנות מתקדם  
תשפ"ב

## אוניברסיטת בן גוריון



### הערות חשובות:

- ייתכן תצטרכו לכתוב פונקציות עזר עבור הפונקציונאליות שהוגדרה לעיל.
- ניתן להניח כי המידע שניתן בקובץ קונפיגורציה תקין לחלוטין, והמספר שמתקבלים כארגומנטים תואמים.
- על מנת לעמוד בדרישות ה ADT הגנרי תצטרכו להוסיף פונקציות נוספות לקבצי headers וsource.
- אין ליצור הקצאת זיכרון עבור אובייקט X מסויים מחוץ לפונקצייה ייעודית "הוספת X".
- אין לנקות הקצרת זיכרון עבור אובייקט X מסויים מחוץ לפונקצייה ייעודית "מחיקת X".

### ○ בשני ההערות הקודמות הכוונה לstructs שקיימים לכם במערכת.

### יצירת מערך של משתנים פרימיטיבים או מערך של פוינטרים מכל

### סוג שהוא ניתן לבצע לא דרך פונקצייה ייעודית.

- יש לבדוק לאחר כל הקצאת זיכרון האם המקום הוקצה כראוי. במידה ולא יש לשחרר את כל המקומות שהוקצו לטובת אותו תהליך, במידה וקיים.
- במידה ומתקבלים מצביעים בפונקציות אפשר להניח כי הערכים בפונקציות נכונים ואין צורך לבדוק תקינותם (לדוגמא, מצביע לשם של פוקימון לא יתקבל מצביע למספר). אולם, בהחלט מצופה ממכם לבדוק האם המצביעים לא מצביעים ריקים. במידה והמצביע ריק, מצופה שהפונקצייה תפסיק את עבודתה ותחזיר null או סטטוס של כישלון. (בהתאם למבנה הפונקצייה)
- בכל מקום שלא צויין אחרת אפשר להניח קלט תקין.
- חשבו היטב איפה אתם ממקמים שמירה של structs וכיצד אתם מפרידים בין קבצי ה ל c. חלוקה נכונה של אחריות וחלוקה נכונה של מודלים ו design נכון הם חלק חשוב מחלוקת הציון.
- כל מודול צריך לעמוד בפני עצמו בלי קשר לשימוש ספציפי כזה או אחר של מודול אחר.
- אין להשתמש במבני נתונים מורכבים (מערך זה בסדר) מסוג כלשהו מעבר למה שהוגדר לכם.
- בזמן קליטה של מידע מהמשתמש, ניתן להניח buffer של 300 תווים על מנת לקלוט את המידע מהמשתמש. אין לשמור (!) במערכת גודל קבוע או להניח גודל מירבי 300. במערכת ישמר רק מידע כגודל המידע שבזמן ריצה קיבלנו מהמשתמש.
- בקריאה מקובץ הקונפיגורציה ניתן להניח גם buffer של עד 300 תווים – לשורה בקובץ.





הפקולטה למדעי ההנדסה

המחלקה להנדסת מערכות מידע

372-1-2102 תכנות מתקדם

**תשפ"ב**

אוניברסיטת בן גוריון



שאריות מהעבודה הקודמת:

אין לשמור בmain את הפוקימונים בשום צורה חוץ מאשר בעזרת המבנה נתונים

.BattleByCategory

אתם מגישים את מודול הפוקימון – כל פונקציונליות שלא קשורה למודל הזה לא צריכה

להיות שם. במיוחד לא שאריות מהmain של העבודה הראשונה.

אין צורך ליצור ממודול הפוקימון ADT !

### דליפות זיכרון

העבודה תיבדק לזיהוי דליפות זיכרון (memory leaks). וודאו כי אתם מוחקים את כל

הערכים שיש למחוק, ולא משאירים זיכרון ללא מצביע אליו בשום שלב של התוכנית.

היעזרו בתוכנה [valgrind](#) בכדי לזהות דליפות זיכרון (כפי שמתואר [כאן](#)).

### הגדרות נוספות:

עליכם ליצור קובץ הגדרות Defs.h שלו תוסיפו את ההגדרות הבאות, והשתמשו בעבודתכם

בהן (אתם חייבים להוסיף את כל הגדרות אלו! ניתן להוסיף עוד הגדרות במידת הצורך):

```
typedef enum e_bool { false,true } bool;
```

```
typedef enum e_status { success, failure} status;
```

\*אתם מוזמנים להוסיף עוד סטטוסים על מנת שהעבודה תהיה יותר אינפורמטיבית – אולם

יש להשאיר את סטטוס ההצלחה כsuccess במידה והפעולה התבצעה בהצלחה.

שימו לב כי הפעם אתם חייבים להשתמש בזה על מנת לתמוך בפונקציונליות של headern

שקיבלתם.

```
typedef void * element;
```

```
typedef element(*copyFunction) (element);
```

```
typedef status(*freeFunction) (element);
```

```
typedef status(*printFunction) (element);
```

```
typedef int(*equalFunction) (element, element);//1 first bigger, 0 - same, -1
```

```
second bigger
```

```
typedef char*(*getCategoryFunction)(element);
```

```
typedef int(*getAttackFunction)(element firstElem ,element secondElem ,int*
```

```
attackFirst,int* attackSecond); // return the first attack - second attack . also
```

```
produce the attack of each element, The function transmits the information
```

```
through the pointers received in the function.
```







## הפקולטה למדעי ההנדסה

המחלקה להנדסת מערכות מידע

372-1-2102 תכנות מתקדם

**תשפ"ב**

## אוניברסיטת בן גוריון



### דו"ח – הסבר על העבודה:

בעבודה זו מלבד הבדיקה האוטומטית, אנחנו רוצים לבחון את הארכיטקטורה של הקוד שלכם וכן את כתיבת הקוד שלכם ותכנון הקוד שלכם. לשם כך, בנוסף להגשה של הקוד, אתם מתבקשים להגיש דו"ח מקיף שמכיל את כל המידע שאתם רוצים שנדע על הקוד שלכם, על מנת שבדוק התרגילים שלנו, כאשר יעבור על הקוד – יוכל לקרוא את ההסבר שלכם על הבחירות שלכם בכתיבת הקוד. יש להוסיף עבור כל מידע שאתם מפרטים עליו את הקטע קוד המתאים ובאיזה קובץ הוא נמצא. בודק העבודות יעבור על העבודה שלכם במקביל לקוד עצמו – אך למען נוחות חשוב כי יהיו קטעי קוד בדוח בהתאם להסברים שלכם. דאגו שהקוד שלכם יהיה מתועד בצורה מלאה ונוחה לקריאה והבנה!

### **מבנה הדוח:**

1. מה הארכיטקטורה של הקוד שלכם באופן כללי? איזה מודולים יש לכם בעבודה ומה תחום האחריות של כל מודל.
2. צרפו את ה Makefile שכתבתם לעבודה זו.
3. איך התמודדתם בקוד עם בדיקת מצבים בהם ההקצאה הדינאמית לא הצליחה? יש לצרף דוגמאות קוד.
4. עבור ה ADT של רשימה וערימה:
  - a. יש לצרף את קובץ ה header בשלמותו
  - b. הסבירו את החלוקה שלכם בין הקוד שכתוב בקובץ ה header לבין הקוד שכתוב בקובץ ה C.
  - c. איפה הסטרוקט יושב בקוד ולמה + את קוד הסטרוקט עצמו
  - d. הערכה של גודל הסטרוקט + הסבר
  - e. האם אתם חושבים שכתבתם אותו בצורה החסכונית ביותר?
  - f. איזה פונקציה יוצרת אותו ואיזה הורסת אותו + הסבר על הפעולות שהפונקציות מבצעות.
5. עבור ADT של קרבות לפי קטגוריות:
  - a. איפה הסטרוקט יושב בקוד ולמה + את קוד הסטרוקט עצמו
  - b. הערכה של גודל הסטרוקט + הסבר
  - c. יש לצרף תרשים מלא של האופן שבו בחרתם לממש את האובייקט הנל – על התרשים להיות ברור ומפורט. יש לצרף בנוסף הסבר מילולי על האופן שבו השתמשתם במבני הנתונים האחרים (רשימה וערימה) בקוד של ה ADT הנל.
  - d. הסבירו את האופן שבו ביצעתם את פונקציית היצירה (`createBattleByCategory`) של ה ADT. הוסיפו את הקוד עצמו של





## הפקולטה למדעי ההנדסה

המחלקה להנדסת מערכות מידע

372-1-2102 תכנות מתקדם

### תשפ"ב

## אוניברסיטת בן גוריון



- הפונקציה למסמך והסבירו איך ביצעתם אותה כולל בקריאה לפונקציות חיצוניות לאיזה מודול הן שייכות.
- e. איזה התאמות ביצעתם ובאיזה קבצים, על מנת לאפשר לADT של הקרבות להשתמש בADT של הרשימה והערימה?
- f. הסבירו את האלגוריתם מאחורי הפונקציה fight, הוסיפו דוגמאות קוד במידת הצורך.
- g. האם יש לכם פונקציות פרטיות במודול? (כאלו שמופיעות רק בקובץ C) במידה וכן, צרפו את חתימותיהן והסבירו מה הן מבצעות.
6. עבור מודול הפוקימון:
- a. האם היו פונקציונליות שנאלצתם למחוק כ"שאריות מהעבודה הקודמת?"
- b. האם הוספתם פונקציות למודול? במידה וכן, הסבירו וצרפו את החתימות המתאימות.
7. עבור קובץ main:
- a. איך שמרתם את האובייקטים שלכם בקובץ ה main כדי לבצע את כל הפונקציונליות של העבודה?
- b. תארו את תהליך הקריאה מתוך הקובץ – איך התמודדתם עם המידע שהתקבל? תארו והסבירו מה ביצעתם על מנת שבקוד עצמו תמיד יהיה כמה שיותר יעיל ולא יחזיק בזיכרון מידע גדול יותר מאשר התקבל בפועל על ידי המשתמש
- c. תארו את החלוקה של הקוד שלכם בmain, האם כתבתם הכל תחת אותה פונקציה? האם חילקתם לתתי פונקציות? הסבירו ותנו דוגמאות.
8. סיכום:
- a. האם אתם חושבים שההפרדה הלוגית שלכם בקוד היא המוצלחת ביותר שיכלתם לעשות?
- b. במידה ונחליף את הmain לקובץ אחר, האם יהיו שינויים במודולים אחרים שאתם תצטרכו לבצע? אם כן איזה, ומדוע המצב כך?
- c. מידע נוסף שתמצאו להגיד – פה זה המקום שלכם להוסיף, אם תרצו, מידע שאתם מרגישים שאתם רוצים להגיד על הכתיבה ותהליך הדיבוג שלכם שאתם רוצים שבודק העבודות ידע.





הפקולטה למדעי ההנדסה

המחלקה להנדסת מערכות מידע

372-1-2102 תכנות מתקדם

**תשפ"ב**

אוניברסיטת בן גוריון



### הנחיות הגשה

ההגשה ביחידים בלבד. עבודה שתוגש בפורמט אחר, לא תיבדק.

יש להגיש את הקבצים הבאים:

Pokemon.h

Pokemon.c

(שני קבצי החסות Pokemon יכללו את הפתרון של עבודה 2 – אתם יכולים לשנות את קובץ זה

במהלך העבודה השלישית)

LinkedList.h

LinkedList.c

MaxHeap.h

MaxHeap.c

BattleByCategory.c (אתם לא מגישים את קובץ ה)

Defs.h

– MainSecond.c

פונקציית הmain שלכם כמוגדרת. (כולל קליטת ערכים מהמשתמש וכן יצירת

מסגרת עבור המערכת) זכרו כי מדובר במופע ספציפי שמשתמש במודולים האחרים.

### **קובץ makefile -**

שימו לב : את העבודה אתם יכולים לכתוב בכל סביבת עבודה שנוחה לכם, אך יש

**להגיש** קובץ makefile שכתבתם בעצמכם ללא עזרה משום סביבת עבודה כזו או

אחרת. שם התוכנית להרצה לאחר שימוש בקובץ הינו PokemonsBattles.





## הפקולטה למדעי ההנדסה

המחלקה להנדסת מערכות מידע

372-1-2102 תכנות מתקדם

**תשפ"ב**

## אוניברסיטת בן גוריון



יש להגיש את העבודה על פי הוראות ההגשה שנתנו בנפרד.

את העבודות יש להגיש לאתר הקורס,

בפעילות שתיקרא "Assignments 3 – C second - Submission"

בנוסף, יש להגיש דוח על פי המפורט בעבודה **לתיבת הגשה "הגשת דוחות – עבודה 3"**.  
יש להגיש את הדוח בפורמט של קובץ word בלבד, כאשר שם הקובץ הוא **תעודת הזהות**  
של המגיש.

התרגיל יעבור גם בדיקה אוטומטית וגם בדיקה ידנית. כדי שהתוכנית תעבור בשלום את  
הבדיקה האוטומטית, אין לשנות את השמות של הקבצים, **ואין לשנות את הפורמט של**  
**ההדפסות שניתנו** – הן מבחינת כמות enters, upper/lower case ורווחים. אם ההדפסה  
אינה מדוייקת כנדרש, הבדיקה תיכשל.



התרגיל יעבור בדיקות קפדניות למציאת העתקות, וסטודנטים שימצאו  
כי העתיקו יענשו בחומרה. אנא המנעו מהעתקות.

שאלות לגבי העבודה ניתן יהיה לשאול בפורום מיוחד שיפתח באתר הקורס.

**בהצלחה, ! catch 'em all**

**Before the  
assignment**



**After the  
assignment**

