



# אוניברסיטת בן גוריון

המחלקה להנדסת מערכות מידע

372-1-2102 תכנית מתקדם



**תרגיל בית מס' 3: תאരיך הגשה: 24.12.25 בשעה : 23:59**

## הקדמה

תרגיל זה הוא החלק השני של מערכת ניהול פוקימון. בחלק הראשון יצרתם את ה pokedex, וכעת, בחלק השני נתמך ביצירת קרבנות בין פוקימונים בעזרת בניית ADT גנרי. בתרגיל זה תתבקשו ליצור ADT גנרי לפי הדרישות, שאינו תלוי בשימוש שייעשה בו בהמשך. בנוסף יהיה עליו להשתמש במשק ה-ADT כדי ליצור את מערכת הקרבנות שתפורט בעבודה הבאה. أنا לפני תחילת פתרת התרגיל קראו עד הסוף את העבודה על מנת למנוע אי הבנות.

שלום שוב! זה אני, פרופסור C, הזמן לא דיברנו. כל הכבוד על העבודה הטובה על הפוקדים. מכיוון שעזרתם לי מאוד, אני רוצה לנצל את היכולות שלכם פעם נוספת. המציאות הבאה עליה אני עובד היא אכן ללא אمان. אני מעוניין ליצור מכוני פוקימונים שלא מאמנים על מנת ליעל את השוק. למי מכם שלא בקיא בתחום, מכון הינו מקום בו נervalים קרבנות פוקימונים. המשימה שלכם היא ליצור מכון אשר יודע באופן מיידי לשולף את הפוקימון בעל הסיכויים הכי גבוהים לניצח בכל קרב. כמו במשימה הקודמת אוחזים מהרווחים לא תקבלו, אבל ציון טוב ותהיilit עולם על הפרה. בהצלחה.

לモטיבציה: [https://www.youtube.com/watch?v=Oc\\_ifA4lGJo](https://www.youtube.com/watch?v=Oc_ifA4lGJo)





## הפקולטה למדעי ההנדסה

המחלקה להנדסת מערכות מידע

372-1-2102 תכנית מתקדם



## אוניברסיטת בן גוריון

בחלק זה נתמקד בכתיבה נכונה של קוד, חלוקה למודלים וכתיבה של ADT גנרי. העבודה זו תבוצע בשני שלבים: בשלב הראשון עליהם ליצור ADT גנרי לפי הדרישות, שאינם תלויים בשימוש שייעשה בו בהמשך.

בשלב השני, עליהם להשתמש במשחק ה-ADT שיצרתם כדי ליצור את מערכת הkersות של הפוק'ונים שלהם.

לפני שתתחלו 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 של המערכת.

במידה ויש סטייה כלשהי בין הכתוב בעבודה זו לבין המידע בקובץ הרצה – יש להסתמך על הריצה שסופה בקובץ.

**בעבודה זו תמשכו שלושה מבני נתונים גנרים שונים.**

מבנה הנתונים אותו תמשכו הינם רשיימה מקוורת גנרית, ערמת מקסימום ומבנה נתונים של קרסות לפי סוגים. (אוטו נפרט בהמשך).

בשלושת מבני הנתונים, אתם בוחרים את המימוש שלהם בפועל. כמובן שיש לכתוב את הקוד בצורה הטובה ביותר והיעילה ביותר שניתן. כמו כן, ניתן להניח כי המשתמש שירצה להשתמש במודול שלנו יעשה שימוש נבון וישלח פונקציות תקיןות. אין צורך לבדוק באיזהishi צורה או לדאוג לגבי מקרה שפונקציה של המשתמש שקיבלוינו תגרור לשגיאה במודול שלנו. בכל הפונקציות מוגדר לכם חלק מהחתימה ומה הפונקציה מבצעת.

אין להוסיף פונקציות נוספות למודול מעבר לפונקציות שניתנו לכם בהוראות העבודה.

### :LinkedList

תמשכו ADT של רשיימה מקוורת גנרית, המוגדרת כך: על הרשימה המקוורת לתמוך בשמרות ערך עבור כל חוליה. הרשימה תתמוך בפעולות הבאות:

1. יצירה של רשיימה גנרית (`createLinkedList`) מקבלת כל המידע אודות האובייקט

הגנרי שתשמור, תאתחל רשיימה ותחזיר מצביע לרשיימה.

2. הריסה של רשיימה גנרית (`destroyList`) אחראית למחיקה של הרשימה, ופינוי הדיכרון.

3. הוספה של חוליה לרשיימה (`appendNode`) פונקציה אשר מקבלת אלמנט שנרצה להוסיף, ומוסיפה אותו כחוליה לסוף הרשימה.





## הפקולטה למדעי ההנדסה

המחלקה להנדסת מערכות מידע

372-1-2102 תכנית מתקדם



## אוניברסיטת בן גוריון

4. מחיקה של חוליה מהרשימה (`deleteNode`) פועלה אשר מקבלת אלמנט ששמור כחוליה בראשימה (יש לוודא כי שמור, במידה ולא יעודכן הסטטוס של הפעולה ככישלון) ומוחקמת את החוליה שבה שמור מהרשימה, ומנקה את הזיכרון שלו.
5. הדפסת הרשימה (`displayList`) הפעולה מדפיסה את כל האיברים בראשימה.
6. חיפוש בעזרת מפתח (`searchByKeyInList`) הפעולה מקבלת מצביע למידע המשמש מפתח חיפוש – מידע מסוים (אלמנט) שבעזרתו מוחפשים בראשימה.
- הfonקציה עוברת על הרשימה ובודקת, בעזרת פונקציית עזר שהתקבלה מהמשתמש בעת ייצירת המבנה, האם הערך השמור בכל חוליה "מתאים" למפתח שהתקבל. אם נמצאת חוליה מתאימה – יוחזר הערך המלא השמור באותה חוליה. אם לא נמצאה חוליה מתאימה – יוחזר NULL. אם קיימים כמה ערכים מתאים, יוחזר הראשון שנמצא בראשימה (כלומר זה שנכנס ראשון).
- רמז ודוגמה: כדי למשוך סעיף זה, יש לבקש מהמשתמש ביצירת המבנה פונקציה שתבדוק את הקשר בין אלמנט לבין מפתח החיפוש.
- לדוגמה, אם האלמנט השמור הוא מספר שלם (זהו), אפשר להגיד קритריון שבו "אלמנט מתאים למפתח אם המספר מתחילה במפתח". במקרה זה, אם כל מספר בראשימה האם הוא מתחילה ב-2. הפונקציה תחזיר את הערך הראשון בראשימה שבערו התנאי מתקיים.

### :MaxHeap

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

1. ייצרת ערימה (`createHeap`) פועלה אשר מקבלת את כל המידע אודות האובייקט הגנרי שיישמר. בנוסף מקבל את המספר המקסימלי של איברים בערימה, ואת השם הייחודי של הערימה. הפונקציה תיצור את המבנה נתונים כנדרש.





## הפקולטה למדעי ההנדסה

המחלקה להנדסת מערכות מידע

372-1-2102 תכנות מתקדם



## אוניברסיטת בן גוריון

2. העתקת ערימה (copyHeap) הfonקציה מקבלת מצביע לערימה, ויצרת העתקה عمוקה של ערימה זו. הfonקציה מחזירה אובייקט חדש בזיכרון, שהוא העתק של הערימה שניתנה, למשתמש.
3. הריסה של ערימה (destroyHeap) הפעולה אחראית למחיקה של הערימה, פינוי הזיכרון של כל האיברים בערימה.
4. הכנסת איבר לערימה (insertToHeap) הfonקציה מקבלת מצביע לאלמנט ובמידה ובערימה אין איברים ככמota המכסה האפשרית, תכניס את האיבר לערימה. יש להכניס את האיבר על פי חוקי הערימה.
5. הדפסת הערימה (printHeap) מדפסה את כל האברים על פי סדרם מהגדול לקטן. על הדפסה להיות בפורמט הבא:

\*name\*:

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

- שימושו לבוטני הדפסה של האיברים הינם על פי החלטתו של המשתמש.  
כלומר: שם הערימה, לאחריו נקודותים וירידת שורה. לאחר מכן יודפס מספר סידורי נקודה ולאחריו רווח. לאחר הרוח יודפס הפרטים של האלמנט. וכך הלאה.  
(אין פה ירידת שורה כלל. המשתמש מחייב אין יודפס עבורה המידע, וכן תהיה ירידת שורה, או אפילו שתיים, רק במקרה שהמשתמש בחר בהדפסת האלמנט שלו יהיה ירידת שורה)

במידה ואין איברים בערימה, יודפס השורות הבאות:

\*name\*:

No elements.

יודפס שם הערימה, לאחריו נקודותים וירידת שורה. לאחר הדפסת השורה של הערימה הריקה יבואו שתי ירידות שורה (כלומר חוץ).

"No elements."

6. החזרת ערך מקסימלי והוצאתו (PopMaxHeap) הfonקציה תחזיר את האיבר המקסימלי בערימה ותמחק אותו מהערימה. במידה והערימה ריקה, יוחזר NULL.
7. החזרת ערך מקסימלי (TopMaxHeap) הfonקציה תחזיר מצביע לאיבר המקסימלי בערימה. במידה והערימה ריקה, יוחזר NULL.
8. החזרת שם הערימה (getHeapId).





## הפקולטה למדעי ההנדסה

המחלקה להנדסת מערכות מידע

372-1-2102 תכנות מתקדם



## אוניברסיטת בן גוריון

9. החזרת מספר האיברים הקיימים כרגע בתוך הערימה. (`getHeapCurrentSize`)  
• יש צורך ליצור פונקציות עזר פנימיות של `heapify` על מנת לשמר על המבנה הנוכחי של מבנה הנתונים.

### :BattleByCategory

אתם מקבלים בעבודה זו את קובץ `H` של אובייקט זה ותצרכו למש את הפונקציות **לפי החתימות שתקבלו**. המלצה – עברו על הוראות ביחד עם החתימות של הפונקציות. שימו לב – אין לשנות דבר(!) בקובץ זה. אתם יכולים לא מגישים קובץ זה, ולכן במידה ותבצעו בו שינויים הקוד שלכם לא יתאפשר. השקיעו מחשבה בחזבה **design** בכך עבור מבנה נתונים זה שייהי **יעיל** ככל הניתן **וישתמש** באופן נבון **במבנה הנתונים האחרים** שבניתם. עלייכם למש **ADT** של קרובות לפי קטגוריות, המבנה יהיה גנרי ויוגדר באופן הבא:  
על המבנה נתונים לתרום בשמירה של אובייקטים על פי קטגוריות.  
כל אובייקט שרצה להשתמש במבנה נתונים זה חייב להכיל קטgorיה שתתואר על ידי שרשרת TOKENS, ומתקפה לקרב שיאופיין על ידי מספר שלהם.  
למבנה יש מכסה לכמות האובייקטים הנשמרים מכל קטgorיה. כלומר ביצירה של קרובות יתקבל מספר (לדוגמא 20) מספר זה אומר שבכל קטgorיה יכולים להיות עד 20 אובייקטים.  
בנוסף כאשר יוצרים קרובות משתמש יכנס את מספר הקטגוריותשהאובייקטים שלו מחולקים על פיהם ובנוסף שורה של שמות הקטגוריות מופרדות על ידי פסיקים.  
כלומר "cat1,cat2,cat3".

מבנה הנתונים מאפשר לשמר את האובייקטים לפי קטגוריות, מאפשר הדפסה למספר של כל האובייקטים, הוספה של אובייקט חדש מסוים ומחיקה של אובייקט חזק ביותר מסוים. בנוסף בהינתן אובייקט חדש, מערכת הקרובות תאפשר לבצע קרב בין אותו אובייקט לאובייקט שיש לו הסיכוי הגדול ביותר לניצח בקרוב בין האובייקטים השמורים המערכת. לוגיקת הקרב תפורט בהמשך.

ב**ADT** זה ניתן להשתמש במערך רק של משתנים פרימיטיבים (`(int, char, double)` או `(int*,char*,double*)`).

עבור שאר המידע שלכם יש להשתמש במבנה נתונים הגנרים שתכתבם קודם.

אתם חייבים **להשתמש** במבנה נתונים שכבר כתבתם – הרשימה המקורית וכן **ערימת מקסימום על מנת למש את מבנה זה. אין להשתמש במידע "פנימי" של האובייקטים אלא רק במידע החשוב בעזרת קובץ `header.h`.**





## הפקולטה למדעי ההנדסה

המחלקה להנדסת מערכות מידע

372-1-2102 תכנית מתקדם



## אוניברסיטת בן גוריון

הגנרי יתמור בפעולות הבאות:

1. ייצירה של קרבנות לפי קטגוריות (createBattleByCategory) פועלה אשר מקבלת את כל המידע אודות האובייקטים שישמרו במערכת. בנוסף, הפעולה מקבלת מספר הקטגוריות וכן את שמות הקטגוריות בשורה מופרדת על ידי פסיקים. נספ על כך הפונקציה תקבל את מספר הקיבולת של האובייקטים מכל קטgorיה. הפונקציה תיצור את האובייקט ותחזיר מצביע אליו.
2. הריסת קרבנות לפי קטגוריות (destroyBattleByCategory) פועלה תקבל מצביע למבנה נתונים ותהייה אחראית למחיקה של המבנה, כל האובייקטים השמורים בו ופנוי הזיכרון.
3. הוספה של אובייקט למערכת – (insertObject) הפעולה מקבלת מצביע למבנה הנתונים ומצביע לאובייקט אותו יש להוסיף למערכת. הפעולה תקבל אובייקט בעל קטgorיה. במידה והקטgorיה מופיעה כבר באובייקט הקרבנות, האובייקט יתווסף למערכת. במידה והקטgorיה לא קיימת, הוא לא יתווסף ויתקבל סטטוס כישלון. במידה ומספר האובייקטים עבור אותה קטgorיה הגיע למכסה האובייקט לא יתווסף גם כן, ויתקבל סטטוס כישלון.
4. הדפסה של כל האיברים באובייקט הקרבנות (displayObjectsByCategories) פועלה מקבלת מצביע לאובייקט הקרבנות ומדפסה את כל האובייקטים ששמורים לפי קטgorיות. האובייקטים יודפסו באופן מסודר מהחזק ביותר לחלש ביותר. למשל, בהינתן 3 קטgorיות cat1,cat2,cat3 כאשר לקטgorיה 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 בקטgorיה מסוימת (removeMaxByCategory) פועלה מקבלת מצביע למבנה הקרבנות ושם של קטgorיה. הפעולה מוחקת את האובייקט החזק ביותר מאותו קטgorיה (במידה וקיים) ומחזירה אותו למשתמש. במידה ולא קיים, יוחזר NULL.
6. החזרה של מספר האובייקטים בקטgorיה מסוימת (getNumberOfObjectsInCategory) פועלה מקבלת מצביע למבנה הקרבנות





## הפקולטה למדעי ההנדסה

המחלקה להנדסת מערכות מידע

372-1-2102 תכנית מתקדם

## אוניברסיטת בן גוריון



שם של קטgorיה. הפעולה מחזירה כמה אובייקטים שמורים במבנה מאותו  
קטgorיה שהתקבל.

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 זהה שניצח את הקרב בצורה הטובה ביותר. בקרב יש

להתחשב בקטgorיות של כל אחד מהאובייקטים, זאת מכיוון שלפי הקטgorיה עשויה

להשתנות המתקפה. כמובן יש לחשאל 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 ומתקפה זהה בין

האובייקט הנבחר לאובייקט המתkeletal לקרב, יהיה תיקו.

במידה ולקרב יש צורך לבצע שובר שיוון בין שני פוקימונים או יותר בעלי מתקפה

זהה לקרב מסוים יבחר לקרב הפוקימון מהקטgorיה שכותבה קודם בראשימת

הקטgorיות שמתקבלת ביצירה של האובייקט (כלומר categories שמתקובל

ב createBattleByCategory ).

אם לא קיימים אובייקטים שנייתן להלחם נגדו יוחזר NULL. אחרת יוחזר המנצח.

במידה והיה תיקו, יוחזר האובייקט שנבחר מתוך המערכת.

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

יבחר האובייקט מהקטgorיה שנכנסה ראשונה בראשימת קטgorיות ביצירת ADT.

ניתן להניח כי לא יהיו שני אובייקטים באוטה קטgorיה בעלי אותה מתקפה.

על הפונקציה להציג 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\*





## הפקולטה למדעי ההנדסה

המחלקה להנדסת מערכות מידע

372-1-2102 תכנית מתקדם

## אוניברסיטת בן גוריון



פירוט הדפסות:

"The final battle between:\n"

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

"against "

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

"THE WINNER IS:\n"

במקרה של תיקו, ההדפסה תראה بصورة 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\*

IT IS A DRAW.

פירוט הדפסות:

"The final battle between:\n"

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

"against "

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

"IT IS A DRAW.\n"

בפונקציה זו המטרה היא לנצח את הקרב. על כן, נדרש למצוא את האובייקט לו הסיכוי הטוב ביותר לנצח ובפרט הגודל ביותר. כולם – נבחר את האובייקט עם הפרש הגובה ביותר מהאובייקט שמולו בקרב (שהוא כמובן משתנה בהתאם ליריב והקטגוריות). בקובץ ההרצאה לדוגמה שעולה עם העבודה יש המונחים של קרבנות - יש להסתכל עליהם כדי למצוא עוד מקרים שונים. כולל מקרים "מיוחדים" שהייה ברור בצורה הטובה ביותר ביוטר אין עובדת המערכת.





**מבנה נתונים זה (BattleByCategory)** **צריך לעמוד באופן עצמאי ללא קשר לפוקימונים.** כל משתמש יוכל לבחור איזה אובייקטים יהיו בו. אתם כמשתמשים תשתמשו בו בהמשך כדי לבצע את הפעולות עבורי אובייקטים של פוקימונים.

**תפריט ראשי:** (שםו לב כי החומר הוא ה"משתמש" (האפליקציה) שלנו בכל המודולים  
ומבני הנთונים שיצרנו – יש לעשות הפרדה בין כל מודול כאשר כתובים את הקוד)

## הרצאה של התוכנית:

עליכם לבנות קובץ PokemonsBattleCenter.c שיסתמש בכל הפקנציות שתוארו לעיל.

**התוכנית עצמה תרוץ בעזרת הפקודה:**

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

כאר

- **numberOfTypes** הינו מספר שלם אשר מציין את כמות סוגי הפקים מונחים אותם

נבדוק, ואוותם יש לקרוא מן הקובץ (לדוגמא 3)

הינו מספר שלם אשר מתאר את כמות הפוקיונים המקסימלי capacityInType -

שניינן להכניס למערכת הקרוות בכל אחד מהסוגים השונים (לדוגמא 20)

- configurationFile הינו נתיב לקובץ קונפיגורציה של סוג פוקימונים ופוקימונים

אותם יש לקרוא מtower הקובץ ולהכנסם למערכת. (אotto דבר מהעובדת הקודמת)

\*שים לב כי הפעם – לא מתקבל מספר הפוקימונים שמופיעים בקובץ

הkonfigورציה!

הkonfigورציה!





## הפקולטה למדעי ההנדסה

המחלקה להנדסת מערכות מידע

372-1-2102 תכנות מתקדם



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

את תחילת הסוגים יש לזהות לפי הcotract Types. אחריה יופיעו רשימה של הסוגים על פי  
הכמות שניתנה בשורת הרצה. בין השורות פרידו 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 Water Against Fire אפקטיבי נגד בקרב, לא בהכרח יהיה כתוב  
אצל Water שהוא אפקטיבי בקרב Against Fire .  
אין מניעה כי המידע יופיע בשתי שורות נפרדות. כלומר:

Type1 effective-against-me:Type2

Type1 effective-against-me:Type3

הסוג שופיע בתחילת השורה הינו סוג המקור, במידה והשורה מסוג -

**against**, הסוגים ברשימה הינם אפקטיבים נגד סוג המקור בקרב. במידה והשורה מסוג

**me**, המקור אפקטיבי נגד סוגים אלו בקרב.

לאחר המידע הנ"ל שלא בהכרח יופיע אודות כל הסוגים, יופיע הcotract Pokemons שכתוב  
ומתחתיו כל הפוקימונים שיש להכניס למערכת על פי הפורמט שנכתב. אין רוחים כלל בין  
 הפרטיהם והם מופרדים בפסיקים.

ניתן להניח את תקינות המידע בקובץ הקונפיגורציה.





הפקולטה למדעי ההנדסה

המחלקה להנדסת מערכות מידע

372-1-2102 תכנות מתקדם



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

שם של פוקימון או סוג תמיד יהיה מילה אחת ללא רווחים. כנל לגבי זה הפוקימון. קובץ הקונפיגורציה שינתן לכמ' יהיה **תיקן לוגית**. כלומר, לא ינתנו פוקימונים עם סוג שלא הוגדר. לא יוגדרו פוקימונים מסוימים מעל המכסה שמותר להחזיק. לא יהיה סוג של פוקימון פעמיים וכו'. בנוסף, ניתן להניח כי לא יהיה בקובץ קונפיגורציה כמוות של פוקימונים מסוימים מעל המכסה המותרת לסוג.





## הפקולטה למדעי ההנדסה

המחלקה להנדסת מערכות מידע

372-1-2102 תכנית מתקדם

## אוניברסיטת בן גוריון



### Main - פונקציונליות:

כאשר התוכנית מתחילה היא תחילת קוראת מן הקובץ את כל המידע שפורט – במידע בקובץ הוא מסד הנתונים שלנו. **חשוב: את הפוקימונים יש לאחסן בעדרת אובייקט ה BattleByCategory בלבד(!).** לאחר מכן יוצג למשתמש הודעה 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 בדיקן ח'.

התוכנית תחכה לקלט מהמשתמש ולאחריו אנטר. הקלדת tuo שונה מהאפשרויות הנל תגרור הודעה 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





**הפקולטה למדעי ההנדסה**

**המחלקה להנדסת מערכות מידע**

**372-1-2102 תכנית מתקדם**

**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

פירוט הדפסות:

"Please choose a valid number.\n"

פונקציונליות הפקודות:

(1) – מדפיס את כל המידע אודות הפוקימונים שנמצאים במערכת הקרבנות. לאחר מכן יוצג שוב התפריט. סדר הדפסה יהיה לפי.types השונים, ועל פי הסדר של אותם טיפים בקובץ הקונפיגורציה. הפוקימונים יודפסו בהתאם לקטגוריה של.type על פי חזק המתקפה שלהם. הראשון יהיה הפוקימון החזק ביותר הבא אחריו שני החזק ביותר וכן הלאה. פורמט הדפסה כפי שמופיע 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:





## הפקולטה למדעי ההנדסה

המחלקה להנדסת מערכות מידע

372-1-2102 תכנות מתקדם



PuppyPokemon, Fire Type.

Height: 0.70 m Weight: 19.00 kg Attack: 70

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.

שנמצאים במערכת הקרבנות. 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 קטgorיה צריך לשתנות בהתאם.

פורמט ההדפסה כפ' שמופרט בפונקציית הדפסה של סוג פוקימון.





**הפקולטה למדעי ההנדסה**

**המחלקה להנדסת מערכות מידע**

**372-1-2102 תכנית מתקדם**



לדוגמה:

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

-Insert Pokemon to battles training camp (3 Pokimmon נוסף אל הפוקימונים השמורים במערכת הקרבנות. ניתן להניח תקינות קלט של המידע אודוט הפוקימון שנכנס! . לאחר בחירה באפשרות זו המערכת תדפיס למסך הודעה המבקשת לדעת איזה סוג של פוקימון תרצה להכניס למערכת. במידה והסוג אינו קיים תוצג הודעה שגיאה והתפריט יודפס בשנית. במידה והוא קיים המערכת תבקש להכניס את כל המידע שהכרחי ליצירת פוקימון. לאחר מכן המערכת תכניס את הפוקימון למערכת הקרבנות 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.

לדוגמה:

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:

**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

אוניברסיטת בן גוריון



Height: 0.50 m Weight: 5.40 kg Attack: 50

פירוט הדרישות 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"

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

תדפיס הודעה המבקשת להזין סוג של פוקימון ממנו רוצים למחוק את הפוקימון חזק ביותר מהמערכת. במידה והסוג אינו קיים תוצג הודעה שגיאה. במידה והסוג קיים במערכת אך אין פוקימונים מסווג זה, תוצג הודעה שגיאה. במידה וישנם פוקימונים מסווג זה, המערכת תוציא ממערכת הקרבנות את הפוקימון החזק ביותר ותדפיס למסך מי נמחק.

לדוגמה:

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 תכנית מתקדם

## 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 ההוראות 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, עליהם לאפשר ייצרת פוקימון אליו תרצו להלחם נגד אחד הפוקימונים השמורים במערכת הקרבנות. ניתן להניח תקינות קלט של המידע אודות הפוקימון שנכנסו! . לאחר בחירה באפשרות זו המערכת תדפיס למסך הודעה המבקשת לדעת איזה סוג של פוקימון תרצה להכניס למערכת. במידה והסוג אינו קיים תוצג הודעה שגיאה והתפריט יודפס בשנית. במידה והוא קיים המערכת תבקש להכניס את כל המידע שהכרחי לייצרת פוקימון.

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

לאחר ייצרת הפוקימון המערכת תבחר את הפוקימון שנגדו יערך הקרב מבין הפוקימונים השמורים במערכת, תנעה מולו קרב ותדפיס מידע אודות הקרב. במידה ואין פוקימונים במערכת הקרבנות תוצג הודעה שגיאה שלא ניתן לבצע את הקרב. • לסוגים של הפוקימונים יש משמעות בשלב הקרבנות. המתקפה של כל פוקימון עלולה להשנות בהתאם לפוקימון נגדו הוא נלחם. להזכירם, לכל סוג פוקימון יש רשיימה





**אוניברסיטת בן גוריון**

**המחלקה להנדסת מערכות מידע**

**372-1-2102 תכנית מתקדם**



של סוגים אשר אפקטיביים נגדו בקרב (effective\_against\_me), ורשיימה של סוגים אשר סוג המקור אפקטיבי נגדם בקרב (effective\_against\_others). בנוסף אין סימטריה בהכרח בין הסוגים. למשל אם בראשימה של סוג Fire כתוב כי Water אפקטיבי נגדו בקרב, לא בהכרח שברשימה של Water נגד מי הוא אפקטיבי בקרב "מצא".

יש לחשב את המתקפה של כל פוקימון לפי האם הפוקימון שמולו אפקטיבי נגדו בקרב. במידה וכן, יפחתו 10 נקודות מהמתקפה שלו. למשל, במידה ויש פוקימון מסוג Fire עם מתקפה של 50, ופוקימון של Water עם מתקפה של 40, ידוע כי Water אפקטיבי נגד Fire בקרב, יפחת מהפוקימון מסוג Fire 10 נקודות מהמתקפה. למשל במקום 50 המתקפה שלו תהיה 40. במקרה זה, לשני הפוקימונים תהיה מתקפה של 40 ולכל יהיה תיקו בקרב. לא ירד לפוקימון יותר מ10 נקודות, במידה וברשימה של סוג א של פוקימונים שאפקטיביים נגדו (נגד סוג א) יופיע סוג ב או שברשימה של סוג ב בראשימה של הסוגים שהוא (סוג ב) אפקטיבי נגדו יופיע סוג א ירד לפוקימון מסוג א 10 נקודות. גם אם הוא מופיע בשתי הרשימות הללו ירד רק 10 נקודות. המתקפה לא יכולה להיות מספר שלילי. וכך בעקבות ההפחתה המספר יהיה שלילי, יש להתיחס אליו כאמור.

לדוגמה:

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:

**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.





**הפקולטה למדעי ההנדסה**

**המחלקה להנדסת מערכות מידע**

**372-1-2102 תכנות מתקדם**

**אוניברסיטת בן גוריון**



Height: 1.70 m Weight: 90.50 kg Attack: 90

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:





**הפקולטה למדעי ההנדסה**

**המחלקה להנדסת מערכות מידע**

**372-1-2102 תכנית מתקדם**



**9.9**

Please enter Pokemon attack:

**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





הפקולטה למדעי ההנדסה

המחלקה להנדסת מערכות מידע

372-1-2102 תכנית מתקדם



6 : Exit

5

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





הפקולטה למדעי ההנדסה

המחלקה להנדסת מערכות מידע

372-1-2102 תכנית מתקדם

אוניברסיטת בן גוריון



IT IS A DRAW.

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.

פירוט ההזדמנויות האפשריות:

"Please enter Pokemon type name:\n"





**הפקולטה למדעי ההנדסה**

**המחלקה להנדסת מערכות מידע**

**372-1-2102 תכנות מתקדם**



"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"

(6) Exit - יציאה מסודרת מהמערכת – כולל מחיקה וניקוי הזיכרון. לאחר ביצוע הnikioon

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 הדפסה:

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





## הפקולטה למדעי ההנדסה

המחלקה להנדסת מערכות מידע

372-1-2102 תכנות מתקדם



## אוניברסיטת בן גוריון

### הערות חשובות:

- יתכן שתצטרכו לכתוב פונקציות עזר עבור הפונקציונליות שהוגדרה לעיל.
- ניתן להניח כי המידע שניtin בקובץ קונפיגורציה תקין לחלוtin, והמספר שמתקבלים כארגומנטים תואמים.
- במקרה שבמהלך התוכנית בניסיון לשימוש בהקצת זכרון דינמי, יהיה כשלון, על החומר להציג למשתמש את הודעה הבאה:  
"ח.ל.No memory available."
- לאחר מכן יש לנוקוט את הזיכרון שהוקצה במערכת בצורה הטובה ביותר ולצאת מהתוכנית.
- אין ליצור הקצת זיכרון עבור אובייקט X מסוים מחוץ לפונקציה "יעודית" "הוספה X".
- אין לנוקוט הקצת זיכרון עבור אובייקט X מסוים מחוץ לפונקציה "יעודית" "מחיקת X".
- בשתי הערות הקודמות נכונה structs שכיחים לכם במערכת. יצירת מערך של משתנים פרימיטיבים או מערך של פוינטרים מכל סוג שהוא ניתן לבצע לא דרך פונקציה "יעודית".**
- יש לבדוק לאחר כל הקצת זיכרון האם המקום הוקצה כראוי. במידה ולא יש לשחרר את כל המיקומות שהוקטו לטובות אותו תהליך, במידה וקיים.
- במידה ומתקבלים מצביעים בפונקציות אפשר להניח כי הערכים בפונקציות נוכנים ואין צורך לבדוק תקינותם (לדוגמה, מצביע לשם של פוקימון לא יתקבל מצביע במספר). אולם, בהחלט מצופה מכם לבדוק האם המצביעים לא מצביעים ריקים. במידה וה המצביע ריק, מצופה שהפונקציה תפסיק את עבودתה ותחזיר null או סטטוס של כישלון. (בהתאם לבנייה (הפונקציה)
- בכל מקום שלא צוין אחרת אפשר להניח קלט תקין.
- חשבו היטב איפה אתם ממקמים שמיירה של structs וכיידם מפרידים בין קבצי ה C.
- חלוקת נכונה של אחריות וחולקה נכונה של מודלים וdesign נכוון הם חלק חשוב מחלוקת הציון.
- כל מודול צריך לעמוד בפני עצמו בלי קשר לשימוש ספציפי צזה או אחר של מודול אחר.
- אין להשתמש בבני נטוניים מורכבים (מערך זה בסדר) מסוג כלשהו מעבר למאה שהוגדר לכם.
- בזמן קליטה של מידע מהמשתמש, ניתן להניח buffer של 300 תאים (299 תווים ו0) על מנת לקלוט את המידע מהמשתמש. אין לשמר(!) במערכת גודל קבוע או להניח גודל מירבי 300. במערכת ישמר רק מידע כגודל הזמן שבו ריצה קיבלנו מהמשתמש.





## הפקולטה למדעי ההנדסה

המחלקה להנדסת מערכות מידע

372-1-2102 תכנות מתקדם

## אוניברסיטת בן גוריון



- בקריאה מקובץ הkonfigورציה ניתן להניח גם buffer של עד 300 תאים (299 תווים ו0) – לשורה בקובץ.

### SHARES מהעבודה הקודמת:

אין לשמר בחום את הפוקימונים בשום צורה חז' מאשר בעזרת המבנה נתונים .BattleByCategory

אתם מגישים את מודול הפוקימון – כל פונקציונליות שלא קשורה למודול זהה לא צריכה להיות שם. במיוחד לא SHARE מהחומרן של העבודה הראשונה.

**אין צורך ליצור מודול הפוקימון ADT !!**

### דלייפות זיכרון

העבודה תיבדק לזרוי דלייפות זיכרון (memory leaks). וודאו כי אתם מוחקים את כל הערכים שיש למוחק, ולא משאירים זיכרון ללא מצביע אליו בשום שלב של התוכנית. היעזרו בתוכנה [valgrind](#) כדי לzechot דלייפות זיכרון (כפי שמתואר [כאן](#)).





## הפקולטה למדעי ההנדסה

המחלקה להנדסת מערכות מידע

372-1-2102 תכנית מתקדם



## אוניברסיטת גוריון

### הגדרות נוספות:

עליכם ליצור קובץ הגדרות.h. Defs.h שלו תוסיפו את הגדרות 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 סטטים על מנת שהעובדת תהיה יותר אינפורטיבית – אולם יש להשאיר את סטטוס ההצלחה success במידה והפעולה הטעינה בהצלחה. שימוש לב Ci הפעם אתם חייבים להשתמש זהה על מנת לתמוך בפונקציונליות של headers. שקיבלתם.

```
typedef void * element;  
typedef element(*copyFunction) (element);  
typedef status(*freeFunction) (element);  
typedef status(*printFunction) (element);
```

```
/* equalFunction :Compares two elements:  
 * returns: 1 if first > second,  
 *          0 if they are equal,  
 *         -1 if first < second. */  
typedef int(*equalFunction) (element, element);
```

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

```
/* getAttackFunction :Calculates the attack of both elements.  
 * Returns (attackFirst - attackSecond) and stores each attack value in the  
 given pointers. */  
typedef int(*getAttackFunction)(element firstElem ,element secondElem ,int*  
attackFirst,int* attackSecond);
```





## הפקולטה למדעי ההנדסה

המחלקה להנדסת מערכות מידע

372-1-2102 תכנית מתקדם

## אוניברסיטת גוריון



### הנחיות הגשה

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

יש להגיש את הקבצים הבאים:

Pokemon.h

Pokemon.c

(שני קבצי Pokemon יכללו את הפתרון של עבודה 2 – אתם יכולים לשנות את קובץ זה במהלך העבודה השלישית)

LinkedList.h

LinkedList.c

MaxHeap.h

MaxHeap.c

작 (קובץ H של מודול זה אתם לא מגייסים. הוא ניתן לכם עם העבודה ואנחנו נצף אותו לעבודה שלכם במבנה בצורה אוטומטית)

Defs.h

– PokemonsBattleCenter.c

פונקציית החום שלכם כמוגדרת. (כולל קליטת ערכים מהמשתמש וכן יצירת מסגרת עבור המערכת) זכרו כי מדובר במופע ספציפי שמשתמש במודולים האחרים.

### - makefile קובץ

את העבודה אתם יכולים לכתוב בכל סביבת עבודה שנוחה לכם, אך יש להגיש קובץ makefile שכתבתם בעצמכם ללא עזרה משום סביבת עבודה צזו או אחרת – כפי שלמדנו בקורס ולא שימוש במקררו בקובץ.

שם התוכנית להריצה לאחר שימוש בקובץ הינו **PokemonsBattles**.





## הפקולטה למדעי ההנדסה

המחלקה להנדסת מערכות מידע

372-1-2102 תכנית מתקדם



## אוניברסיטת בן גוריון

יש להגיש את העבודה על פי הוראות ההגשה שנטנו בנפרד.  
את העבודות יש להגיש לאתר הקורס,  
בפניות שתיקרא "Assignments 3 – C second - Submission".

התרגיל יעבור גם בדיקה אוטומטית **וגם בדיקה ידנית של הקוד עצמו**. כדי שהתוכנית תעבור בשלום את הבדיקה האוטומטית, אין לשנות את השמות של הקבצים, **ואין לשנות את הפורתט של הדפסות שניתנו** – הן מבחינת\_cmoots enters,upper/lower case ורוחחים. אם הדפסה אינה מדוייקת כנדרש, הבדיקה תיכשל.

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



שאלות לגבי העבודה ניתן להזמין לשאול בפורום מיוחד שייפתח באתר הקורס.

**בהצלחה, I catch 'em all**

**Before the assignment      After the assignment**

