

## תרגיל בית 1 מת"מ – חלק יבש

נעם רוזה 313599573

כרמל דוד 207174327

### amount set – א

חילקנו את הבעיה לשני אובייקטים שונים:

1. Amount Set

2. Element Node

```
typedef struct ElementNode_t *ElementNode;
struct ElementNode_t {
    ASElement element;
    double amount;
    ElementNode next_node;
};
```

```
struct AmountSet_t {
    ElementNode first_node;
    ElementNode iterator;
    int size;
    CopyASElement copyElement;
    FreeASElement freeElement;
    CompareASElements compareElements;
};
```

Amount Set ממומש על ידי רשימה מקושרת, כאשר כל איבר ברשימה המקושרת הוא מסוג Element Node. כל Element Node מכיל בתוכו אובייקט element, מסוג void\*, אליו המשתמש יכול להזין את סוג האלמנט שבו יבחר להשתמש.

שימוש בפונקציות:

- Copy element – בעת הכנסה של אלמנט חדש לרשימה, נשתמש בפונקציית copy שסופקה על ידי המשתמש כדי להעתיק את האלמנט הנתון לתוך הרשימה
- Free element – במחיקה של אלמנט נפנה לפונקציה זו כדי למחוק את המידע הפנימי שהמשתמש הגדיר.
- Compare element – נשתמש בפונקציה זו כדי למצוא את המיקום בו יש להכניס איבר חדש, או כדי למצוא את האיבר שאלינו להסיר או לשנות את כמותו.

### Matamazom – א

חילקנו את הבעיה לשלושה אובייקטים שונים, כאשר שניים מהם מכילים אובייקט נוסף בתוכם:

#### 1. Matamazom

```
struct Matamazom_t {
    AmountSet storage;
    List orders;
    unsigned int number_of_orders;
};
```

האובייקט הראשי שלנו. מכיל מחסן הממומש על ידי Amount Set, רשימת הזמנות הממומשת על ידי List, וinti שסופר את מספר ההזמנות שיצרנו.

#### 2. Storage

```
typedef struct Product_t {
    char *name;
    unsigned int product_id;
    MatamazomAmountType amountType;
    MtmProductData customData;
    double sales;
    MtmCopyData copyData;
    MtmFreeData freeData;
    MtmGetProductPrice prodPrice;
} *Product;
```

אובייקט הממומש כ Amount Set ומכיל בתוכו את כל המוצרים במחסן ואת הכמות שלהם.  
אובייקט Product מסוג struct מכיל את כל הנתונים שיוסופקו על ידי המשתמש עבור כל מוצר.

### Orders .3

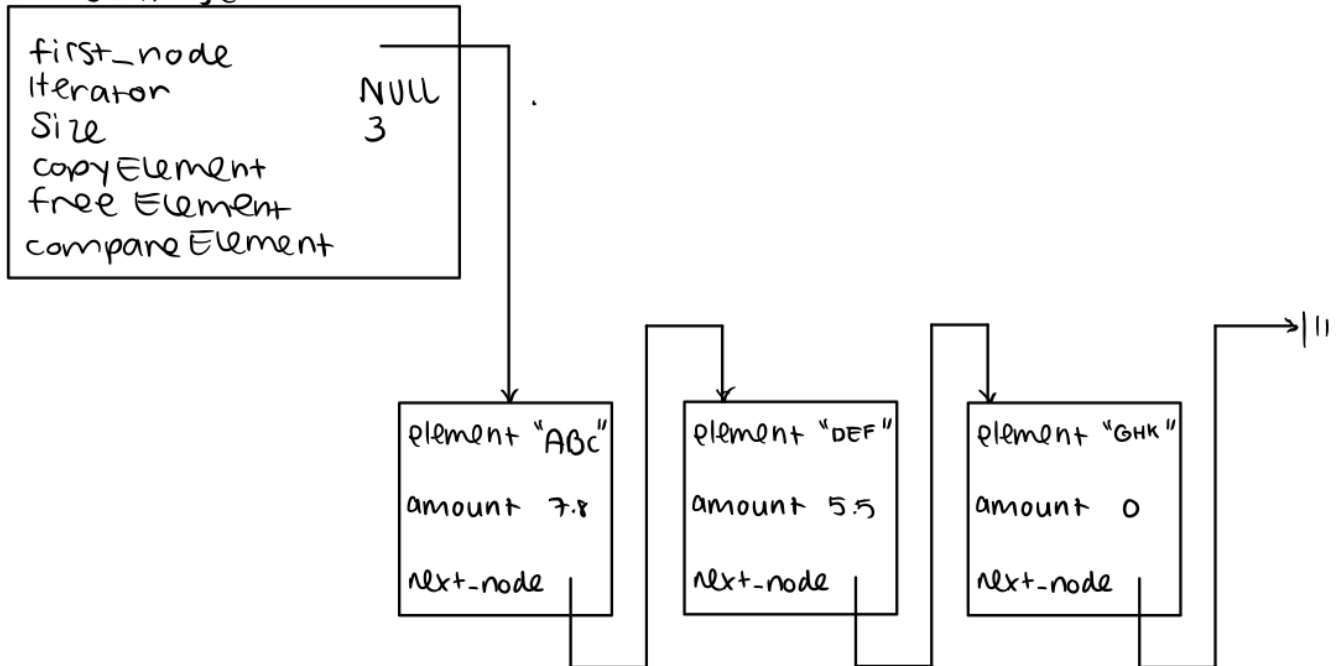
```
typedef struct Order_t {
    unsigned int order_id;
    AmountSet products_in_order;
} *Order;
```

אובייקט מסוג List שמכיל את כל ההזמנות שנעשו.  
תחת כל הזמנה יש Amount Set שמכיל רשימה מקושרת המכילה את כל המוצרים שקיימים תחת אותה הזמנה.

שימוש בפונקציות:

- FreeOrder – בשילוח הזמנה, נרצה לאחר עדכון הכמויות והמכירות במחסן למחוק את ההזמנה עצמה. נשתמש בפונקציה זו כדי למחוק את ההזמנה ואת כל המוצרים תחתיה.
- CopyOrder – ביצירת הזמנה חדשה, נבנה node של ההזמנה החדשה, ואותו נעתיק בעזרת CopyOrder לתוך הרשימה שלנו.

Amount Set:



במצב זה ה Amount Set שלנו מכיל שלושה איברים ברשימה מקושרת.  
 כל איבר מצביע לאיבר הבא אחריו, והאיבר האחרון מצביע לכתובת ריקה.  
 גודל ה Amount Set הוא 3, והאיטרטור מוגדר ל NULL כי לא הוגדר אחרת.  
 ה Amount Set מכיל גם פונקציות שיאפשרו לנו להעתיק, להשוות ולמחוק את האלמנטים (במקרה זה מוג \*char).

Matamatzom:

AmountSet storage  
 List orders  
 Unsigned int number-of-orders

מתאמזום הוא האובייקט הכללי שבתוכו יש מחסן (storage) שממומש Amount Set והזמנות שממומשות כlists.

בנוסף, יש מספר שלם חיובי שמייצג את מספר ההזמנה אשר מוצמד להזמנה בעת יצירתה וגדל ב-1 בין הזמנה להזמנה.

Storage: (AmountSet - מניין)

ASElement Element  $\Rightarrow$  product (מבנה struct)

double amount  $\Rightarrow$  double amount (כמות המוצר במחיר)

ASElement next\_node  $\Rightarrow$  next\_prod (מבנה struct המאגד את המוצר הבא)

המחשן מכיל Element Nodes כאשר כל אחד מייצג את המוצר ואת הכמות שלו במחשן. הproduct הוא מצביע לstruct אשר מכיל בתוכו את כל המידע על המוצר אשר מסופק ע"י המשתמש וכן מכיל את הפונקציות להעסקה ומחיקה של customData.

Product

char\* name

unsigned int product\_id

MatamAmazonAmountType amountType

MtmProductData customdata

double sales

MtmCopyData copyData

MtmFreeData freeData

MtmProductPrice prodprice

ההזמנות ימומשו כאלמנטים בתוך רשימה מקושרת, כל הזמנה תכיל מספר הזמנה ייחודי ו Amount Set של מוצרים בתוך ההזמנה. בתוך Element Node יישמר הproduct עצמו וכן כמות מהמוצר בהזמנה הספציפית.

List Orders:

Order (הnode הראשון)

unsigned int order\_id

AmountSet products-in-order

Product product  
double amount