



final project- pizzeria

מגישות:

אור אברמוביץ ת"ז 209373067

אורי כץ ת"ז 314632449

1. הקדמה קצרה על המערכת

בנינו מערכת לניהול פיצרייה.

המשתמש יטען פיצרייה קיימת מתוך קובץ טקסט או בינארי, עם שם ועם כתובת ויוכל לבצע הזמנות.

לפיצרייה יש רשימת חברי מועדון אשר יכולים להשתמש ולצבור נקודות.

יש רשימת קופונים אשר מספקים הנחה נוספת.

נוסף על כך יש תפריט אשר בו לכל פיצה מחיר שונה בהתאם לגודל הפיצה, לסוג הבצק ולתוספות.

פירוט מלא על כל תכונות המערכת החל מעמוד הבא.

2. הסבר על כל מבנה

:Pizzeria

```
typedef struct
{
    AllCoupons allCoupons;
    Menu menu;
    MemberManager membersArr;
    int numOfMembers;
    Order* orders;
    int numOfOrders;
    char* name;
    Address address;
    eOrderSortType typeSort;
}Pizzeria;
```

מבנה פיצרייה הוא המבנה הראשי של המערכת.

במבנה זה נמצאים כל נתוני הפיצרייה: שם הפיצרייה, כתובתה, תפריט המסעדה, מערך קודי הקופון, מערך חברי המועדון ומערך ההזמנות כאשר על כל אחד מהם קיים פירוט בהמשך.

המשתמש טוען את הפיצרייה מתוך קובץ טקסט או בינארי.

הוא יכול להוסיף הזמנות לפיצרייה, למיין את ההזמנות לפי שם הלקוח, לפי מחיר ההזמנה או לפי תאריך ההזמנה, ולמצוא הזמנה במערך לפי אותם פרמטרים.

פיצרייה מכילה משתנה מסוג AllCoupons – שמחזיק את מערך הקופונים של הפיצרייה.

משתנה מסוג Menu – שישמור את תפריט המסעדה, הוא בעצם מכיל רשימה של פיצות.

MemberManager - שמחזיק את מערך חברי המועדון של הפיצרייה .

Order – מערך של כל ההזמנות של הפיצרייה

int – משתנה שיחזיק את מספר ההזמנות הקיימות לפיצרייה.

char* – מחרוזת בגודל לא ידוע שתחזיק את שם הפיצרייה.

Address – כתובת הפיצרייה.

typeSort – משתנה מסוג enum אשר מחזיק את הפרמטר על פיו נבצע מיון של מערך ההזמנות, על פי שם הלקוח, תאריך ההזמנה או מחיר ההזמנה.

:Order

```
typedef struct
{
    Date      date;
    Time      time;
    eOrderType typeOfOrder;
    Address    address;
    LIST      pizzaOrderList;
    Coupon     coupon;
    Customer   customer;
    int        member; //1 - if the customer is a club member 0 - if not
    double     priceToPay;
}Order;
```

מבנה order הוא מבנה שמנהל את נתוני הזמנה.

נעדכן בו את כל פרטי ההזמנה, תאריך ושעה, את פרטי הלקוח, ואם הוא חבר מועדון, נציין זאת.

נעדכן את סוג ההזמנה, ואם מדובר במשלוח עד הבית, נעדכן גם את הכתובת אליה תישלח ההזמנה.

הלקוח יבחר אילו פיצות מהתפריט הוא מעוניין להזמין, הוא יכול להזמין גם את אותה הפיצה כמה פעמים.

לבסוף נחשב את המחיר הכולל של ההזמנה.

נשאל את הלקוח אם יש לו קוד קופון. אם קוד הקופון קיים במאגר הקופונים, נחשב את המחיר לאחר ההנחה.

אם הלקוח הוא חבר מועדון, תינתן לו האופציה להשתמש בנקודות אשר יזכו אותו בהנחה. נוסף על כך בסיום ההזמנה הוא יצבור נקודות (30% מהמחיר הסופי מעוגל למספר שלם).

לבסוף נעדכן את המחיר הסופי של ההזמנה.

הוא מחזיק משתנה מסוג Date – תאריך ההזמנה, Time – שעת ההזמנה

typeOfOrder – סוג ההזמנה, משתנה מסוג enum אשר מחזיק את סוג ההזמנה, אם מדובר בשיבה במסעדה, לקחת הביתה או משלוח עד הבית.

משתנה מסוג Address – משתנה זה יתעדכן רק אם מדובר במשלוח עד הבית. במקרה זה הכתובת תכיל את הכתובת אליה יישלח המשלוח.

LIST – רשימה של פיצות.

יחזיק את כל הפיצות שהלקוח הזמין בהזמנה ואת פרטיהן.

Coupon – אם ללקוח קיים קופון אשר נמצא במאגר הקופון של הפיצרייה משתנה זה יחזיק אותו.

Customer – משתנה שיחזיק את פרטי הלקוח.

Int – משתנה שערכו 1 אם הלקוח הוא חבר מועדון, 0 אם לא.

Double – משתנה שיאותחל תחילה ל-0 ובסיום ההזמנה ישמור את סכום ההזמנה לאחר ההנחות, כלומר: סכום ההזמנה שנותר לשלם.

:Date

```
typedef struct
{
    int    day;
    int    month;
    int    year;
}Date;
```

מבנה Date הוא מבנה שמחזיק את תאריך ההזמנה, כלומר: התאריך שבו הלקוח מעוניין לקבל את ההזמנה.

נעדכן בו פרטים אלה באמצעות 3 מספרים שלמים (int), 1 day שייצג את היום 2 month מייצג את החודש ו-3 year שייצג את השנה.

נשים לב שהשנה לא יכולה להיות לפני השנה הנוכחית, שנת 2024. כמו כן היום צריך להיות תואם לחודש, מכיוון שיש חודשים שיש בהם 31 ימים, יש כאלה שיש בהם 30 ימים ויש את פברואר עם 28 או 29 ימים. אנו בודקות זאת במחלקה זו.

:Time

```
typedef struct
{
    int    hour;
    int    minute;
}Time;
```

מבנה time הוא מבנה שמחזיק את זמן ההזמנה, כלומר: הזמן שבו הלקוח מעוניין לקבל את ההזמנה.

נעדכן בו פרטים אלה באמצעות 2 מספרים שלמים (int), 1 hour שייצג את השעה ו-2 minute שייצג את הדקות.

נשים לב ששעות פתיחת הפיצרייה הן 12:00-22:00, לכן גם הזמן צריך להיות בטווח זה, אנו בודקות זאת במחלקה זו.

:Address

```
typedef struct
{
    char*   city;
    char*   street;
    int     houseNum;
}Address;
```

מבנה time הוא מבנה שמחזיק את הכתובת.

אנו משתמשים בו בתכנית בשני מקומות:
הראשון, כאשר מעדכנים את כתובת הפיצרייה.
השני, אם בוצעה הזמנה למשלוח עד הבית, נעדכן את הכתובת אליה נשלח את ההזמנה.

נעדכן בו פרטים אלה באמצעות 3 משתנים:
2 מחרוזות בגודל לא ידוע בשביל העיר והרחוב (אין מגבלות על מחרוזות אלה)
ומספר שלם (int) בשביל מספר הבית.

מספר הבית צריך להיות בין 1-100, אנו בודקות זאת במחלקה זו.

:Customer

```
typedef struct
{
    char* name;
    char* id;
    int points;
}Customer;
```

מבנה Customer הוא מבנה שמחזיק את נתוני הלקוח: שמו, ת"ז ונקודות (הנקודות רלוונטיות רק אם הלקוח הוא חבר מועדון).

נתונים אלה נשמרים במשתנים הבאים:

2 מחרוזות בגודל לא ידוע (char*) בשביל שם הלקוח ות"ז הלקוח.

Int שבו נשמור את מספר הנקודות של הלקוח.

תחילה נקלוט את נתוני הלקוח, את שמו, את הת"ז אשר תכלול 9 ספרות ונוסף על כך נאתחל את מספר הנקודות ל-10.

:MemberManager

```
typedef struct
{
    Customer* membersArr;
    int membersCount;
}MemberManager;
```

מבנה זה מכיל רשימה של כל חברי המועדון של הפיצרייה.

נתונים אלה נשמרים במשתנים הבאים:

מערך מסוג Customer אשר מכיל את כל חברי המועדון.

Int שבו נשמור את חברי המועדון הקיימים במערך.

לחבר מועדון יש נקודות שאותן הוא יכול לנצל ולצבור במהלך ההזמנה.

לפני התשלום הוא יישאל אם הוא מעוניין להשתמש בנקודות, מובן שלא יוכל להשתמש במספר נקודות הגדול מסכום ההזמנה או במספר נקודות גדול ממה שקיים ברשותו.

בסיום ההזמנה יחושב ויתעדכן מספר הנקודות שלו על ידי חיסור מספר הנקודות שישתמש וצבירת כמות נקודות כ-30% מסכום ההזמנה.

:Pizza

```
typedef struct
{
    eToppingType topType[MAX_TOP];
    eDoughType    doughType;
    int           size;
    double        price;
}Pizza;
```

מבנה Pizza מכיל את נתוני הפיצה:

גודל הפיצה, סוג הבצק, תוספות הפיצה ומחירה.

המשתמש יבחר 4 תוספות, אחת עבור כל רבע בפיצה.

המשתמש יבחר נתונים אלה, ונחשב בהתאם למחירים קבועים שהגדרנו ב־general את מחיר הפיצה.

בנוסף על כך יצרנו פונקציה מיוחדת אשר מציירת את הפיצה:

גודל הציור יהיה לפי גודל הפיצה שנבחרה, ב־general מוגדרים הגדלים הרלוונטיים.

את הבצק נצייר בעזרת ריבוע כוכביות, בתוכו נחלק את הפיצה לארבעה חלקים ובכל חלק יופיע תו בהתאם לתוספת שנבחרה, גם את התווים הגדרנו ב־general.

נתוני המבנה מוחזקים על ידי כמה משתנים:

מערך מסוג eToppingType בגודל 4 אשר נשמור בו איזו תוספת יש בכל רבע של פיצה, בתא 0 תהיה התוספת השמאלית למעלה, בתא 1 תהיה התוספת הימנית למעלה, בתא 2 השמאלית למטה וב-3 הימנית למטה.

משתנה מסוג eDoughType אשר יחזיק את סוג הבצק שנבחר.

Int שבו נשמור את גודל הפיצה.

doublei שישמור את מחיר הפיצה לאחר שנחשב אותו.

:Menu

```
typedef struct
{
    LIST pizzaList;
}Menu;
```

מבנה זה מחזיק את תפריט המסעדה.

בתחילת התכנית יבנה המשתמש את התפריט על ידי הוספת פיצות.

התפריט בנוי על ידי רשימה List וקיימת פונקציה שמוסיפה פיצות לרשימה זאת.

מובן שאם יש פיצה שמופיעה כבר בתפריט לא יהיה ניתן להכניס אותה בשנית.

כאשר נבצע הזמנה, הלקוח יוכל לבחור פיצות אך ורק מתוך התפריט.

:Coupon

```
#define MAX_LEN 5
typedef struct
{
    char coupon[MAX_LEN+1];
}Coupon;
```

מבנה Coupon הוא מבנה שמכיל מחרוזת בגודל 6 תווים כולל ה-'0' אשר מייצג קופון.

הקופון הוא מחרוזת של 5 תווים כאשר 3 התווים הראשונים הם אותיות גדולות ולאחר מכן שני התווים האחרונים הם ספרות אשר מייצגות את אחוז ההנחה שקוד הקופון נותן, לדוגמה: ABC20 נותן 20 אחוז הנחה.

:AllCoupons

```
typedef struct
{
    Coupon** arrCoupons;
    int numOfCoupons;
} AllCoupons;
```

מבנה AllCoupons הוא מבנה הכולל את מערך הקופונים הקיימים ברשות הפיצרייה.

בתחילת התכנית, כאשר נבנה את הפיצרייה, המשתמש יוכל להוסיף קופונים למאגר.

כאשר נבצע הזמנה נשאל את הלקוח אם יש לו קופון מסוים. אם כן, נבדוק אם הקופון קיים במאגר הקופונים. אם כן נשמור אותו ובסיום ההזמנה נחשב את ההנחה שהוא נותן.

המבנה מכיל מערך של מצביעים מסוג coupon (Coupon** arrCoupons)

ומספר שלם (int) אשר מחזיר את מספר הקופון במערך זה.

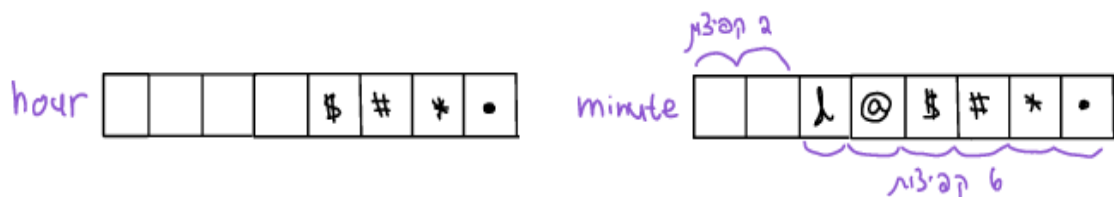
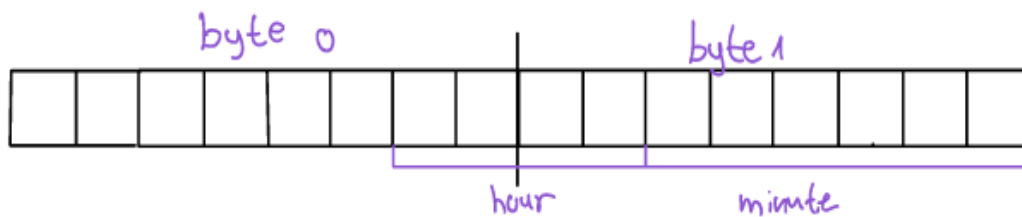
3. דחיסת מבנה

דחיסת מבנה

אנחנו דחסנו לקובץ הנייטאי את המבנה Time.
המבנה מכיל 2 איברים.

* hour int - המספר האפשרי הצדף ביותר הוא 24, אבל נחסם אותו המספר
הוא 22-2. מכיוון שאנו רוצים לאחסן באותו סף, ציונים 4 bits.
* minute int - המספר האפשרי הצדף ביותר הוא 59, לכן ציונים 6 bits.

ביחד: 14 bits, בוגר 2 bytes



$$\text{data}[0] = (\text{pTime} \rightarrow \text{hour} - 12) \ll 4 \mid (\text{pTime} \rightarrow \text{minute} \gg 2)$$

$$\text{data}[1] = \text{pTime} \rightarrow \text{minute} \ll 6$$

ופאשר נקרא מהקובץ:

$$\text{pTime} \rightarrow \text{hour} = ((\text{data}[0] \gg 4) \& 0x0F) + 12;$$

$$\text{pTime} \rightarrow \text{minute} = ((\text{data}[0] \& 0x0F) \ll 2) \mid (\text{data}[1] \gg 6);$$

4. פעולות אפשריות במערכת

- הוספת קופון למאגר הקופונים של הפיצרייה – המשתמש יוכל להכניס למאגר קוד קופון חדש. המערכת תבדוק שקוד הקופון שהוכנס עומד בתנאים (3 אותיות גדולות ו-2 ספרות, והאם קיים כבר במאגר).
- הוספת חבר מועדון למאגר חברי המועדון של הפיצרייה – המשתמש יוכל להכניס חבר מועדון למאגר חבר מועדון חדש. המערכת תבדוק את תקינות תעודת הזהות, והאם הלקוח לא נמצא כבר במאגר. כאשר המערכת קולטת את שם המשתמש – היא מסירה רווחים מיותרים, והופכת כל אות ראשונה במילה לאות גדולה.
- הוספת פיצה חדשה לתפריט – המערכת תציג למשתמש את הגדלים האפשריים לפיצה: SMALL – 10x10, MEDIUM – 16x16, LARGE – 22x22. המשתמש יבחר את הגודל הרצוי ולאחר מכן יבחר את סוג הבצק: רגיל, טבעוני, ללא גלוטן או מקמח מלא. לאחר מכן המשתמש יבחר את התוספות הרצויות בעבור כל אחד מארבעת הרבעים.
- הוספת הזמנה חדשה למאגר ההזמנות – המערכת תבקש מהמשתמש שם ותעודת זהות. כאשר המערכת קולטת את שם המשתמש – היא מסירה רווחים מיותרים, והופכת כל אות ראשונה במילה לאות גדולה. המערכת בודקת באמצעות תעודת הזהות אם הלקוח הוא חבר מועדון, ומדפיסה הודעה בהתאם. המערכת תקלוט מהמשתמש את התאריך והשעה הרצויים, ולאחר מכן תקלוט את סוג ההזמנה – ישיבה במסעדה, טייק אוו או משלוח. אם נבחר משלוח, המשתמש יתבקש להכניס עיר, רחוב ומספר בית (גם כאן מתבצע תיקון למחרוזות בדומה לתיקון שם הלקוח). לאחר מכן המערכת תדפיס את התפריט ותבקש מהמשתמש לבחור את הפיצה הרצויה. היא תשאל האם הלקוח מעוניין בפיצה נוספת. **אחר כן תבדוק האם למשתמש יש קוד קופון, ואם כן, האם אכן קיים במאגר. אם כן תבצע את ההנחה.** אם הלקוח חבר מועדון, המערכת תדפיס את מספר הנקודות של הלקוח ותוריד מסכום ההזמנה את הנקודות לפי בחירתו ובנוסף תחסיר נקודות אלו ממספר הנקודות שלו ובסיום ההזמנה תוסיף לו נקודות כ-30% מסך ההזמנה. המערכת תדפיס את הסכום הסופי לתשלום.
- הדפסת כלל פרטי המערכת: שם הפיצרייה, כתובת, תפריט הפיצרייה, מאגר הקופונים, מאגר חברי המועדון ומאגר ההזמנות.
- הדפסת מאגר הקופונים/מאגר חברי המועדון/תפריט/מאגר ההזמנות בלבד
- מיון מאגר ההזמנות – המערכת תבצע מיון של מאגר ההזמנות לפי מחיר, תאריך ההזמנה או שם המזמין.
- חיפוש במאגר ההזמנות – המערכת תבצע חיפוש במאגר לפי מחיר, תאריך הזמנה או שם המזמין.
- המערכת תשמור את נתוניה לקובץ טקסט וקובץ בינארי.

6. חלוקת אחריות

(השרטוט בעמוד הבא)

התחלנו את הפרויקט ביחד, העלנו את כל הרעיונות ולאחר מכן בנינו ביחד את המבנים הבסיסיים.

לאחר מכן, חילקנו את שאר המבנים בינינו וקבענו כמה ימים אחרי שוב.

סגרנו פינות יחד ואז חילקנו את שאר העבודה כך שאחת עשתה קבצים בינאריים כולל דחיסת המבנה והשנייה עשתה את הקבצי טקסט וה-Macros

5. שרטוט המערכת:

