

מבחן 10010 שאלון X

הוראות כלליות:

1. אין לשנות מבנים, הגדרות פונקציות .
2. יש לממש כל שאלה בקבצים השייכים לה בלבד, אין להוסיף קבצים לפרויקט
3. אין להשתמש במשתנים גלובלים, בפונקציה exit וב goto
4. יש להקפיד לסגור קבצים ולשחרר זיכרון בכל מקום רלוונטי.
5. בנושא bits אין להגדיר מבנה עם : וכמות Bits
6. **יתכן ותדרשו להוסיף include בקבצי השאלות**
7. כל סעיף במבחן נבדק באופן בלתי תלוי בסעיף אחר באותה שאלה. לכן ההמלצה לפתור כל סעיף גם אם לא הצלחתם לפתור את הסעיף שלפניו.
8. כל דבר שלא צוין שאסור אזי מותר

שאלה 1 (40 נקודות) : (קבצים q1.c q1.h)

סעיף א' (25 נקודות):

ממש/י את הפונקציה `createArrayOfStrings` המקבלת כפרמטר כמות לא מוגדרת מראש של מחרוזות, זו אחר זו, כאשר רשימת המחרוזות מסתיימת בערך `NULL`. בנוסף הפונקציה מקבלת משתנים נוספים לפי הגדרתך ובסדר שתחליט, ראה פירוט בהמשך.

עבור כל מחרוזות שהפונקציה קיבלה היא מייצרת העתק שלה במחרוזות חדשה ושומרת את כל המחרוזות במערך אותו היא מחזירה.

בנוסף הפונקציה מחזירה ע"י מצביעים :

1. את מספר המחרוזות המערך
2. את אינדקס המחרוזת הארוכה ביותר
3. את אינדקס המחרוזת בה יש את מספר המילים הרב ביותר. ניתן להניח שאין רווחים בתחילת המחרוזות וסופה, כמו כן שיש רק רווח אחד בין כל מילה ומילה במחרוזות.

סעיף ב' (15 נקודות):

השלם/י את הפונקציה הראשית `Q1()`.

- הפעל את הפונקציה `createArrayOfStrings` עם המחרוזות :

- "This is"
- "A good day"
- "abcdefghijklmnopqrstuvwxyz"

- בעזרת המשתנים שהתקבלו מהפעלת הפונקציה הדפס :

- את כל המחרוזות
- את המחרוזת הארוכה ביותר
- את המחרוזת שיש בה את מספר המילים הרב ביותר.

- שחרר את כל ההקצאות שנוצרו בשאלה 1, כך שלא תהיה זליגת זיכרון.

הפלט עבור המחרוזות שניתנו :

```
----- The Strings -----
This is
A good day
abcdefghijklmnopqrstuvwxyz

----- The Longest String -----
abcdefghijklmnopqrstuvwxyz

----- The most words String -----
A good day
```

שאלה 2 (40 נקודות) : (קבצים q2.c q2.h)

נתון המבנה Order המציין הזמנה.

שדות המבנה : שם ההזמנה (ניתן להניח מילה אחת), ברקוד (מחרוזת – מילה אחת), מחיר.

```
#define LEN 9
```

```
typedef struct
{
    char*   name;
    char    barcode[LEN + 1];
    float   price;
}Order;
```

בקבצים GeneralList.c GeneralList.h נתון הקוד של רשימה מקושרת חד כיוונית השומרת נתון כללי, void* , כפי שפורסם בפתרון תרגיל בית 3. (יש להשתמש בקוד זה ללא שינוי)

רוצים לאפשר שמירת מצביעים ל Order , הזמנה, ברשימה מקושרת חד כיוונית. ולבצע מספר פעולות.

נתונות הפונקציות :

1. *int initOrderFromUser(Order* pOrder)* – מאתחלת נתוני הזמנה מהמשתמש.

2. *int initListOfOrder(LIST* plst)* – מאתחלת רשימה מקושרת של הזמנות.

סעיף א' (15 נקודות):

ממש את הפונקציות הנדרשות שיאפשרו :

- חיפוש הזמנה ברשימה המקושרת על פי הברקוד ע"י שימוש בפונקציה L_find.
- הדפסת כל ההזמנות ברשימה המקושרת ע"י שימוש בפונקציה L_print.
- מחיקת הזמנה מהרשימה המקושרת ע"י שימוש בפונקציה L_delete.

סעיף ב' (25 נקודות):

השלם/י את הפונקציה הראשית Q2() הפונקציה :

הפונקציה מבצעת את הפעולות הבאות :

- מאתחלת את הרשימה ע"י קריאה לפונקציה *initListOfOrder*
- מדפיסה את כל ההזמנות באמצעות הדפסת הרשימה
- מוחקת את ההזמנה האחרונה ברשימה
- מדפיסה את כל ההזמנות באמצעות הדפסת הרשימה
- קולטת מהמשתמש ברקוד של הזמנה לחיפוש.
- מחפשת את ההזמנה על פי הברקוד שנקלט מהמשתמש ברשימה ומדפיסה את ההזמנה שנמצאה במקרה ונמצאה או הודעה שההזמנה לא נמצאה.
- משחררת את כל הקצאות שנוצרו בשאלה 2.

```

----- Q2 -----
How many orders in list
3
Please enter order name, barcode (9 chars) and price
cloths 012345 34.5
Please enter order name, barcode (9 chars) and price
shoes 879200 77
Please enter order name, barcode (9 chars) and price
food 45678C 78

----- The list -----

Order : cloths barcode: 012345 price: 34.50
Order : shoes barcode: 879200 price: 77.00
Order : food barcode: 45678C price: 78.00

----- The list after delete last -----

Order : cloths barcode: 012345 price: 34.50
Order : shoes barcode: 879200 price: 77.00

Enter barcode (9 chars) to search
012345
Order found      Order : cloths barcode: 012345 price: 34.50

```

שאלה 3 (20 נקודות) : (קבצים q3.c q3.h)

```
typedef struct
{
    int bankCode;
    int branch;
    int number;
}AccountInfo;
```

נתון: המבנה המתאר נתוני חשבון בנק:

שדות המבנה: קוד הבנק, מספר סניף ומספר חשבון.

ידוע כי:

קוד הבנק הינו מספר בין 1 ל 7 (3 bits)

מספר הסניף הינו מספר בין 1 ל 500 (9 bits)

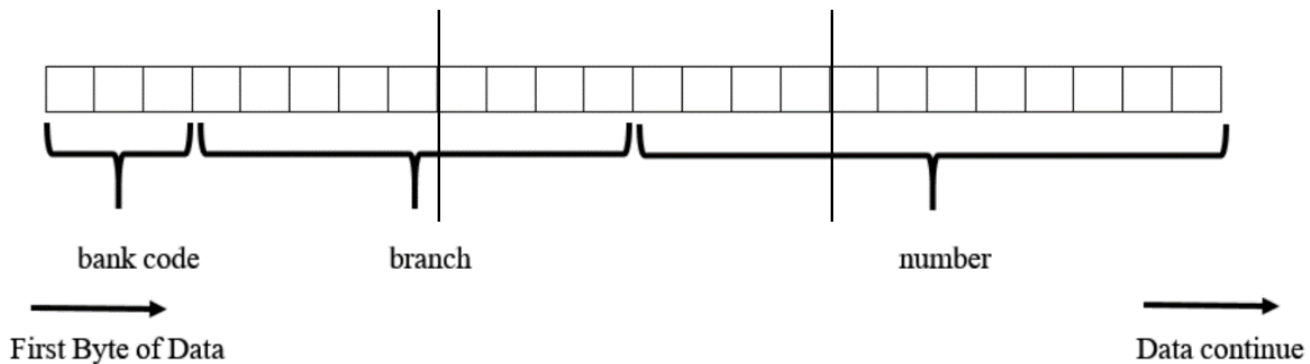
מספר החשבון הינו מספר בין 1 ל 4000 (12 bits)

נתוני החשבונות נשמרים בקובץ בינארי בצורה דחוסה. ידוע שאין יותר מ 255 חשבונות בקובץ הנתון.

פורמט הקובץ הבינארי:

Byte ראשון – מספר החשבונות השמורים בקובץ

כל 3 בתים נתון של חשבון אחד לפי התרשים הבא:



נתון הקובץ בינארי **accounts_comp.bin** בו שמורים הנתונים בצורה דחוסה.

הערה: במידה והנך מוותר על הטיפול בפענוח המיידע בפורמט הדחוס נתון בתיקייה גם הקובץ הבינארי

accounts.bin בו הנתונים לא דחוסים. בחירה באופציה זו תוריד 10 נקודות מציון המבחן.

פורמט הקובץ: מספר החשבונות בקובץ, נתוני החשבונות אחד אחר השני.

ממשי/ את הפונקציה

AccountInfo* createAccountArrFromFile(const char* fileName, int* pCount)

הפונקציה מקבלת את הנתונים הבאים:

filename – שם הקובץ בו שמורים נתוני החשבונות

pCount - מצביע לעדכון מספר החשבונות בקובץ

הפונקציה מייצרת ומחזירה:

מערך של חשבונות על פי הנתונים הרשומים בקובץ.

נתונה הפונקציה הראשית Q3() של התרגיל, הפונקציה :

- קוראת לפונקציה *createAccountArrFromFile* עם קובץ בשם המוגדר בקבוע `FILE_NAME`
- מדפיסה את כל החשבונות
- משחררת את ההקצאות הנדרשות.

נתוני החשבונות בקובץ הנתון :

```
There are 9 accounts
Account: bank 6 branch 400 number 3000
Account: bank 4 branch 350 number 2222
Account: bank 7 branch 120 number 1230
Account: bank 5 branch 80 number 900
Account: bank 3 branch 20 number 567
Account: bank 2 branch 499 number 3500
Account: bank 1 branch 100 number 3020
Account: bank 3 branch 200 number 3400
Account: bank 2 branch 50 number 1277
```