

## תרגיל מספר 1 בשפת C

תשפ"ה סמסטר ב

עליכם לכתוב תכנית המממשת את התפריט הבא:

```
/Users/roeeshahmoon/CLionProjects/Ex1_Stock/cmake-build-debug/Ex1_Stock

=== Stock Management Menu ===
1. Add Stock
2. Print Stocks
3. Double All Stocks Price
4. Drop Stocks Price by x%
5. Find Less Expensive Stock
6. Sort Stocks by Price
7. Sort Stocks Alphabetically
8. Sort Stocks by ASCII Sum of Name
9. Check Palindromic Stock Names
10. Exit
Please enter a number between 0-99:
```

בסיום כל סעיף (מלבד 10) יש להדפיס שנית את התפריט.

### סעיף 0:

קלט תקין לתפריט הוא מספר אי שלילי, חד או דו ספרתי בלבד. אם יודן קלט תקין אשר אינו תואם את אחת מהאפשרויות בתפריט המשתמש יתבקש להזין קלט שוב ושוב עד לקבלת קלט התואם לאופציה מן התפריט. עם זאת, אם יודן קלט אחר שאינו תקין כגון: אותיות, מספרים שאינם בטווח המתאים הוא ייחשב כטעות. לאחר 5 טעויות מצטברות, התוכנית תסתיים בהודעת שגיאה מתאימה. (שימו לב קלט תקין מאפס את מספר הטעויות המצטברות).

### סעיף 1:

המשתמש יכול להוסיף מניה חדשה למערך, לכל מנייה יש שם ומחיר. יש לקלוט את הנתונים אל תוך מבנה struct מתאים, ע"פ הדוגמה המצורפת.

```
Enter stock name: 6006
```

```
Enter stock price: 147
```

```
typedef struct {
    char name[21];
    float price;
} Stock;
```

שם המנייה הינו מחרוזת בעלת 20 אותיות (21 תווים כולל תו סיום מחרוזת). היכולה להכיל אותיות בלבד (a-z או A-Z), כל נתון אחר לא יתקבל.

מחיר מנייה הנקלט בפעם הראשונה הינו מס' שלם וחיובי בלבד. אמנם עבור המשך התוכנית וסעיף 4 בפרט נרצה שיהיה מטיפוס: float.

ניתן לבדוק ע"י הרצת קובץ exe שקיבלתם כיצד לטפל בשגיאות הקלדה.  
לאחר הוספת מנייה ניידה את המשתמש ונדפס הודעה למסך בהתאם.

Stock added.

## סעיף 2:

הדפסת כל המניות הקיימות.  
(יפעל רק אחרי שסעיף 1 הורץ לפחות פעם אחת)

```
1. GOOG - $147.00
2. MSFT - $359.00
3. NVDA - $94.00
4. ESLT - $370.00
5. AMD - $85.00
```

## סעיף 3:

הכפלת הערך עבור כל מחירי המניות והדפסת הודעה מתאימה.  
(יפעל רק אחרי שסעיף 1 הורץ לפחות פעם אחת)

Stock prices have doubled.

## סעיף 4:

הורדת ערך עבור כל מחירי המניות ב-x אחוז, יש לקלוט את x מהמשתמש ע"פ התצורה הבאה:

Enter x%:

יש לוודא שהקלט המתקבל אכן תקין כלומר מספר אי שלילי שלם בין 0-100.  
לסיום יש להדפיס הודעה מתאימה למסך.

Stock prices has dropped by 10%.

(יפעל רק אחרי שסעיף 1 הורץ לפחות פעם אחת)

## סעיף 5:

מציאת מחיר המניה הנמוך ביותר, לסיום יש להדפיס הודעה מתאימה למסך.  
(יפעל רק אחרי שסעיף 1 הורץ לפחות פעם אחת)

Less expensive stock: AMD - \$85.00

## סעיף 6:

מיון המניות לפי מחיר, לסיום יש להדפיס הודעה מתאימה למסך.  
(יפעל רק אחרי שסעיף 1 הורץ לפחות פעם אחת)

Sorted by price.

### סעיף 7:

מיון המניות לפי א-ב האנגלי, לסיום יש להדפיס הודעה מתאימה למסך.  
(יפעל רק אחרי שסעיף 1 הורץ לפחות פעם אחת)

Sorted by name.

### סעיף 8:

מיון כל המחרוזות הקיימות על פי סכום קודי תווי ה ascii של המחרוזת.  
(יפעל רק אחרי שסעיף 1 הורץ לפחות פעם אחת)

Sorted by ASCII sum.

### סעיף 9:

בדוק עבור כל מניה האם שמה הוא פלינדרום.  
(יפעל רק אחרי שסעיף 1 הורץ לפחות פעם אחת)

Palindrome stock: 6006

### פולינדרום – ויקיפדיה

### סעיף 10:

יציאה מהתוכנית.

הנחיות נוספות:

בעבודה זו נגדיר את שני המשתנים בפונקציית התפריט:

```
int stock_count = 0;  
Stock stocks[MAX_STOCKS];
```

כאשר ערכו של MAX\_STOCKS יוגדר ע"י #define וערכו יהיה 10

בעבודה זו חובה להשתמש בפונקציות הבאות, יש לעבוד עם חתימות מדויקות וניתן להעתיקן מדף הוראות זה:

```
void printMenu();
```

פונקציה המדפיסה את התפריט הראשי.

```
void addStock(Stock stocks[], int* ptr_stock_count);
```

פונקציה האחראית על הוספת מנייה, למערך המניות. הפונקציה מקבלת כפרמטרים את מערך המניות ואת כתובת מונה המניות.

(נקודה למחשבה אם היינו מעבירים את מונה המניות כערך רגיל (ולא כמצביע), האם התוכנית הייתה מתנהגת כראוי?)

```
int isValidName(const char* str_buf);
```

פונקציה המקבלת כפרמטר את המידע מהבאפר ובודקת אם שם המנייה תקין.  
במידה והשם תקין תחזיר 1 אחרת תחזיר 0.

```
int isValidPrice(const char* str_buf, float* ptr_price);
```

פונקציה המקבלת כפרמטרים את המידע מהבאפר ובודקת אם מחיר המנייה תקין, מעבירה את הערך התקין בלבד ע"פ הכתובת של מצביע המחיר. במידה והמחיר תקין תחזיר 1 אחרת תחזיר 0.

```
void printStocks(Stock stocks[], int stock_count);
```

פונקציה המדפיסה את המניות הקיימות, הפונקציה מקבלת כפרמטרים את מערך המניות ואת מונה המניות.

```
void doubleStocks(Stock stocks[], int stock_count);
```

פונקציה אשר מכפילה את הערך עבור כל מחירי המניות ומדפיסה הודעה מתאימה. הפונקציה מקבלת כפרמטרים את מערך המניות ואת מונה המניות.

```
void dropStocks(Stock stocks[], int stock_count);
```

פונקציה אשר אחראית על הורדת ערך עבור כל מחירי המניות ב-x אחד, יש לקלוט את x מהמשתמש ולקרוא לפונקציה הנמצאת מתחת אשר מוודא את תקינתו של x. הפונקציה מקבלת כפרמטרים את מערך המניות ואת מונה המניות ולבסוף מדפיסה הודעה מתאימה.

```
int isValidPercent(const char* str_buf, int* ptr_percent);
```

פונקציה המקבלת את המידע מהבאפר ובודקת אם x הינו תקין כלומר מספר אי שלילי, שלם בין 0-100. מעבירה את הערך התקין בלבד ע"פ הכתובת של מצביע האחז. במידה ו-x תקין תחזיר 1 אחרת תחזיר 0.

```
void sortByPrice(Stock stocks[], int stock_count);
```

פונקציה אשר אחראית על מיון המניות ע"פ המחיר.

```
void sortByName(Stock stocks[], int stock_count);
```

פונקציה אשר אחראית על מיון המניות ע"פ א-ב האנגלי.

```
void sortByAsciiSum(Stock stocks[], int stock_count);
```

פונקציה אשר אחראית על מיון כל המחרוזות הקיימות על פי סכום קודי תווי ה ascii של המחרוזות.

```
int asciiSum(const char* str);
```

פונקציית עזר אשר מקבלת מחרוזת קבועה (הפונקציה אינה משנה אותה) ומחזירה את סכום ערכי תווי ה ascii של אותה מחרוזת.

```
void findLessExpensive(Stock stocks[], int stock_count);
```

פונקציה אשר מוצאת את מחיר המניה הנמוך ביותר, לסיום מדפיסה הודעה מתאימה למסך.

```
void findPalindromes(Stock stocks[], int stock_count);
```

פונקציית עזר אשר אחראית על הורדת ערך עבור כל מחירי המניות ב-x אחד, יש לקלוט את x מהמשתמש ולקרוא לפונק' הנמצאת מטה אשר מוודא תקינתו ומדפיסה הודעה מתאימה.

```
int isPalindrome(const char* str);
```

פונקציית עזר אשר מקבלת מחרוזת קבועה (הפונקציה אינה משנה אותה) ומחזירה 1 במידה והמחרוזת הינה פולינדרום, אחרת תחזיר 0.

הערות:

1. ניתן להניח שכל קלט מסתיים בלחיצת enter, מומלץ להשתמש ב fgets().
2. לא ניתן להניח קלטים תקינים, יש לבדוק זאת.
3. קלט מספרי תקין נחשב קלט עם ספרות רצופות, בנוסף המספר צריך להיות בטווח המבוקש, כל קלט אחר יחשב כלא תקין.
4. יש להשתמש בספריות המצוינות למטה בלבד, ללא ספריות נוספות. (שימוש בספריות אחרות עלול להביא לפסילת העבודה)

```
#include <stdio.h>
#include <string.h>
```

5. ניתן להשתמש בחומר הנלמד עד לפרסום העבודה ורק בחומר זה, ובפירוט: תנאים, לולאות, מערכים, מחרוזות, מצביעים ומבנים בלבד.
6. שימו לב להערות על כל סעיף וסעיף, על הפלט להיות זהה לחלוטין לזה של קובץ הריצה.
7. בכל מקום בו מזוהה חריגה בין ההסברים לבין קובץ הריצה יש לידע אותי במייל או בפורום, ובכל מקרה יש לנהוג על פי קובץ הריצה.
8. בכל המיונים שני נתונים בעלי ערך מיון זהה ישמרו על הסדר המקורי שלהם גם לאחר המיון.
9. חובה להשתמש בכל הפונקציות שהוגדרו ע"פ חתימתם הנתונה, ניתן להוסיף פונקציות עזר משלכם.
11. הגשה בזוגות, יש לכתוב מספרי תעודות זהות בהערה בתוך קובץ ההגשה.
12. שם הקובץ יורכב מת"ד של שני הסטודנטים, למשל: 123456789\_123456789.c
13. נא לוודא שהתוכנית עובדת באופן תקין גם בקומפיילר gcc.
14. תוכלו לבדוק את עצמכם ע"י קובץ input.txt וקובץ output.txt שהעליתי לכם למודל לפי דוגמה להרצה בCMD או בטרמינל:

```
gcc ID1_ID2.c -o my_program.exe
my_program.exe < input.txt > output.txt
```

בהצלחה רבה,  
רועי שחמון.