

קורס תכנות 2 - שפת C

תרגיל בית מספר 5

נושאי התרגיל:

- רשימות מקושרות
- עבודה עם קבצים
- קבלת ארגומנטים לתוכנית.

הנחיות כלליות:

1. קראו בעיון את כל ההוראות והקפידו שהתוכניות יפעלו בהתאם לנדרש.

2. בתרגיל זה יש להגיש קובץ אחד, לשאלה אחת.

לקובץ הקוד שמגישים יש לקרוא: (עם מספר ת.ז.):

q1_123456789.c

יש להגיש קובץ c אחד, מכוון כקובץ ZIP. שם הזיפ יהיה EX5_123456789 כאשר 123456789 הוא ת.ז של הסטודנט.

3. כל עוד לא נכתב במפורש אחרת אתם יכולים להשתמש בכל פונקציות ספרייה שהכרנו בהרצאות ותרגולים.

4. ניקוד:

- שאלה 1 - 90%

- נראות - 3%

- תיעוד קוד 7%

5. כמו בכל התרגילים הקודמים גם כאן יש להקפיד לבצע הדפסות / כתיבות לקובץ בדיוק כמו שמופיעות כאן או בקבצי הדוגמא. מכיוון שחלק מהבדיקות הינן אוטומטיות, אי הקפדה עשויה לגרום להורדת נקודות.

6. את התרגיל יש לפתור ולהגיש ביחידים.

7. לא תינתן הארכת זמן. תכננו את הזמן שלכם.

שאלה 1 - CPUs Company

אתם עובדים בחברה בשם TAU- CPUs שמתעסקת בפיתוח מעבדים חזקים. לאחרונה החברה נרכשה בעשר מיליארד דולר ולכן ההזמנות למוצרים גדלו. כתוצאה מריבוי ההזמנות עלתה הצעה לנהל רשימה של המעבדים שתתעדכן בכל יום וכך החברה תוכל לעקוב אחר ההזמנות. אור, חבר הצוות שלכם, ביקש מכם לכתוב תוכנית לניהול מלאי המעבדים של החברה.

הערה חשובה: בתרגיל זה יתקבלו אך ורק פתרונות שמשתמשים ברשימה מקושרת בשביל לאחסן את המעבדים של החברה.

מבנה המעבד:

המעבדים יאוחסנו ברשימה מקושרת, כאשר כל איבר בה מהצורה הבאה:

```
// define struct
typedef struct cpu
{
    Char name [NAME_LENGTH];
    int copies;
    struct cpu *next;
}CPU;
```

כאשר name הוא שם המעבד ו copies הוא מספר המעבדים מסוג זה.

קבועים שחייבים להופיע בתחילת קובץ הקוד שלכם:

```
#define NAME_LENGTH 200
```

כעת במקום להציג תפריט ולקבל הוראות מהמשתמש, נשתמש בקבצים. את רשימת ההוראות תקבלו בקובץ הטקסט **actions.txt** ואת מלאי המעבדים בקובץ **cpus.txt**. את רשימת המעבדים המעודכנת לאחר מילוי ההוראות תדפיסו לתוך הקובץ **updated_cpus.txt**.

לתוכנית שלכם לא יהיה שום ממשק עם המשתמש והקלט והפלט שלכם יהיו רק מתוך ולתוך קבצי טקסט (למעט הדפסות במקרה של שגיאות למיניהן, פירוט בהמשך).

נזכיר, על מנת שהתוכנית תוכל לקבל ארגומנטים כקלט יש להשתמש בחתימה הבאה לפונקציית **main**.

```
Int main(int argc, char* argv[])
```

פעולות בקבצים:

ניתן להניח מראש כי אורך כל שורה לא עולה על 400 תווים.

פענוח ההוראות בקובץ ההוראות

בקובץ ההוראות **actions.txt** כל הוראה תופיע בשורה אחת (בסוף כל הוראה יהיה תו newline, כלומר '\n'). בתחילת הריצה התוכנית תקרא את קובץ ההוראות שורה אחר שורה, תפענח את הפעולה הדרושה ואת הפרמטרים שלה, ותקרא לביצוע הפעולה עם הפרמטרים המתאימים.

קליטת מעבדים מהקובץ לרשימת המעבדים

Initialize

זאת הפעולה הראשונה בקובץ ההוראות והיא צריכה לפענח את קובץ המעבדים **cpus.txt** שורה אחר שורה ולהעביר את המעבד הרלוונטי למבנה המתאים ברשימה המקושרת של מעבדים. בקובץ המעבדים **cpus.txt** כל מעבד יופיע בשורה אחת (בין כל שני מעבדים יפריד '\n') ובה יופיעו השדות לפי הסדר שבו הם אמורים להיקלט במבנה כאשר בין שדות תופיע המחזורת "\$\$\$". דוגמא לשורה שתייצג מעבדים בקובץ:

Cpu1 \$\$\$ 120

Cpu2 \$\$\$ 80

שימו לב, המעבדים לא בהכרח ממוינים לפי סדר השמות.

הדפסת המעבדים המעודכנים לקובץ המעבדים המעודכן

Finalize

זאת הפעולה האחרונה בקובץ ההוראות והיא צריכה לעבור על רשימת המעבדים ולהדפיס מעבד מעבד בקובץ **updated_cpus.txt** לפי הפורמט (שם המעבד, מופרד ב"\$\$\$"), וכמות המעבדים מאותו סוג במלאי). הכתיבה לקובץ **updated_cpus.txt** תהייה בסדר ממוין לפי שם המעבד. לאחר מכן יש לשחרר את כל הזיכרון שהוקצה בתוכנית ולצאת. שימו לב, ה-main בסיום התוכנית יחזיר 0 ומספר שונה מ-0 אם היו בעיות.

קובץ actions.txt

מבנה קובץ actions.txt הוא מהצורה הבאה :

Initialize

...

Returned_from_customer \$\$\$ cpu1 \$\$\$ 12

Fatal_malfunction \$\$\$ cpu2 \$\$\$ 13

(more on orders in the next page)

...

Finalize

שימו לב יש להניח שהקובץ מתחיל תמיד ב Initialize ותמיד מסתיים ב Finalize כמו כן ניתן להניח שמבנה ההוראות קבוע

הפעולות הנתמכות

1. שינוי שם של מעבד

Rename \$\$\$ current cpu name \$\$\$ new cpu name

פעולה זו מקבלת שני פרמטרים בסדר הבא- שם של מעבד קיים והשם החדש של המעבד. הפעולה מחפשת האם קיים מעבד ששמו זהה לשם המעבד שהתקבל. במידה וקיים משנה את שמו, במידה ולא קיים לא עושה דבר. שימו לב, בסיום פעולה זו יש לבדוק שרשימת המעבדים עודנה ממוינת.

2. החזרת מעבד

Returned_from_customer \$\$\$ cpu name \$\$\$ copies

מעבד יכול לחזור למלאי החברה בעקבות לקוח לא מרוצה. החברה תומכת בשירות הלקוחות הטוב ביותר ונותנת ללקוח זמן לא מוגבל להחזרת המוצר. הפעולה תחפש לפי שם המעבד האם הוא קיים במלאי, במידה וקיים יעודכן מספר המעבדים לפי מספר המעבדים שהוחזר מהלקוח. במידה ולא קיים במלאי, יעודכן איבר חדש ברשימת המעבדים לפי הנתונים, כאשר לאחר הוספת המעבד החדש הרשימה תישאר ממוינת לפי שם המעבדים.

`Fatal_malfunction $$$ cpu name $$$ copies`

החברה שלכם מומחית בייצור מעבדים מהאיכות הגבוהה ביותר. בדרך כלל רוב המעבדים לא מתקלקלים לעולם, אך לעיתים מעבדים עלולים להתקלקל. כאשר מעבד מתקלקל החברה משמידה אותו ורווחיה נפגעים. אתם נדרשים לפענח הוראה מסוג זה כדי לעזור לחברה להשתפר באיכות ייצור המוצרים.

פעולה זו מחפשת את המעבד המבוקש לפי שמו. במידה ונמצא המעבד המבוקש יש לבדוק את מספר העותקים שלו. במידה והוא גדול ממספר העותקים הזמין של המעבד, רק מספר העותקים הזמין יושמד מהמלאי, ומספר העותקים של המעבד יעודכן להיות 0. במידה ומספר העותקים המקולקלים קטן ממספר העותקים הזמינים, מספר העותקים הזמינים יעודכן בהתאם. במידה ולא נמצא, לא קורה כלום.

הערות לשאלה 1

- אתם מתבקשים לכתוב את הקוד בקובץ c אחד. מומלץ להשתמש בפונקציות.
- רשימת המעבדים צריכה להיות ממוינת לפי שם המעבד בכל שלב ובפרט כאשר כותבים לקובץ **updated_cpus.txt**.
- שימו לב, יתקבלו אך ורק פתרונות שמשתמשים ברשימה מקושרת בשביל לאחסן את המעבדים.
- אין להקצות מראש זיכרון עבור מספר גדול של איברים. צורת עבודה זו היא צורת עבודה בזבזנית משום שפעולה של הקצאת זיכרון לוקחת הרבה מאוד זמן בהשוואה לפעולות אחרות, ולכן זו גישה לא נכונה. הקצו זיכרון רק כאשר אתם נדרשים לכך.
- שימו לב, השימוש ברשימה מקושרת עוזר לנו לתמוך ברשימת המעבדים, וכך ניתן להוסיף \ למחוק בקלות מעבד מהמלאי.
- התוכנית תקבל את הקלטים שלה מהקומנד ליין. מידע על איך להזין קלט דרך הקומנד ליין מפורט במודל.
- יש במודל דוגמא לקבצי קלט פלט שאתם יכולים להשתמש בהם כדי לבדוק את התוכנית.
- קבצי הקלט הינם : **cpus.txt** , **actions.txt**. קובץ הפלט המעודכן הינו : **updated_cpus.txt**.
- יש לבדוק שמספר הפרמטרים בקלט של הקומנד ליין תקין (שימו לב שהארגומנט הראשון הוא תמיד שם התוכנית ולכן צריכים להיות בסה"כ 4 ארגומנטים) וששני הקבצים הראשונים קיימים. במידה ויש שגיאה בתוכנית עליכם להדפיס הודעת שגיאה מתאימה. את הודעת השגיאה יש להעתיק מקובץ **error_messages** אשר נמצא במודל.
- אפשר להניח שהקלט בקבצים הוא קלט נכון (יינתנו פקודות רק מהסוגים שהוגדרו).
- כל המעבדים ברשימה המקושרת צריכים להיות ממוינים לפי שם המעבד, לכן כל פעם שמוסיפים מעבד חדש הוא צריך להיות מוכנס למיקום המתאים ברשימה.
- כאשר מתווסף מעבד חדש שלא היה קיים קודם לכן יש להקצות זיכרון בשביל לשמור את נתוני מעבד. כלומר בכל שלב כמות הזיכרון המוקצה צריכה להיות בדיוק כמות הזיכרון הדרושה לאכסון המעבדים ולא יותר.
- לאחר כל הקצאת זיכרון, חובה לבדוק האם ההקצאה הצליחה. במידה ולא, יש להדפיס הודעה מתאימה, לשחרר את הזיכרון שהוקצה לפני זה ולצאת מהתוכנית עם סטטוס 1.
- שימו לב, כל התוכניות ייבדקו עם בודק דליפות זיכרון אוטומטי- הקפידו לשחרר את כל הזיכרון שהוקצה בתוכנית לפני היציאה.
- יש לסגור את כל הקבצים שפתחתם לאחר סיימתם להשתמש בהם בעזרת **fclose**.
- אתם יכולים להשתמש בכל ספריה שראיתם בתרגול או בהרצאות.
- תמיד תבדקו האם הקצאת הזיכרון הצליחה או לא.
- כדי לדבג השתמשו בהדפסות משלכם. כאשר אתם מגישים תסירו אותם כדי שהתוכנית שלכם תעמוד בדרישות שהוצגו.

- כל פונקציה צריכה להיות עם דוקומנטציה מהצורה הבאה) יש לכתוב זאת בהערה מעל השימוש בפונקציה):

Input:

return parameter (if the function is void write None.)

Function functionality: (1-3) sentences what is the purpose of this function and how you're going to implement this functionality.

- אסור להוסיף שדות למבנים הנתונים או לממש מבנים אחרים.
- זה הפרויקט הגדול השני שלכם. אנחנו מודעים לקושי התרגיל, מטרתו להכין אתכם לקורסי התכנות הבאים כמו כן למבחן בצורה מיטבית.

חומרי העשרה - רשות

חומרי העשרה על הנושאים השונים שהוצגו בתרגיל בית זה :
מקורות מידע אלו לא יתרמו באופן ישיר לפתרון התרגיל אלא נועדו להרחיב את הידע הכללי שלכם.
שימו לב :

1. קטעי קריאה אלה הם רשות ולא חובה לקרוא אותם .
2. קטעים אלו לא יעזרו לכם לפתור את התרגיל .
3. קראו אותם בזמנכם הפנוי (מי שמעוניין) לאחר שסיימתם את התרגיל .

- https://www.youtube.com/watch?v=cNN_tTXABUA&t=339s
- <https://www.youtube.com/watch?v=lxnlyJYZ6Vw>
- <https://www.computerhope.com/jargon/c/cpu.htm>