# הפרויקט הסופי לקורס שפת C ותקשורת

## הסבר כללי

התרגיל מורכב משני חלקים:

1. תרגיל שבודק ידעת שפת C, אלגוריתמים, מבני נתונים בסיסיים, ניהול זיכרון והרגלי תכנות יעיל
2. חלק שני מבוסס על התרגיל הראשון ומוסיף לו את חלק התקשורת וריבוי משימות (threads) ושימוש בלינוקס

## הנחיות כלליות

1. הפרויקט יפותח במערכת ההפעלה לינוקס
2. הנבחן צריך לקמפל ולהריץ את התוכנית על קבצי קלט שונים, לבדוק מקרי קצה ולטפל בטעויות משתמש או קלט לא תקין
3. התוכנית צריכה להיות יעילה מבחינת זמן ריצה וזיכרון
4. הנבחן יכול להשתמש בכל כלי פיתוח לפי רצונו
5. קבצי קלט יישלחו ביחד עם קוד מקור של התוכנה אחרי השלמת המשימה ובדיקה על ידי הנבחן: התוכנית שלא עוברת קומפילציה או נופלת – ייפסל
6. אפשר להיעזר באינטרנט
7. לא ניתן להשתמש בחבילות עבור הפרוייקט, אלא אם כן נאמר במפורש אחרת.

## חלק ראשון:

לכתוב תוכנית שמנהלת רשימת חייבים למכולת.

התוכנית קוראת קובץ בפורמט CSV (שניתן ליצור בEXCEL). לכל לקוח יש את השדות הבאים:

שם פרטי, שם משפחה, מספר זהות, טלפון וסכום החוב, ותאריך רישום החוב (התאריך צריך להיות מהצורה של: שנים/חודשים/ימים **בלבד** וניתן להניח שיש 12 חודשים שכולם יכולים להכיל 31 ימים. כמו כן כל הנתונים יהיו באנגלית, החוב יכול להיות 0 או שלילי אם הלקוח שילם את חובו או חלק ממנו), מספר זהות מוכרח להיות בן 9 ספרות, מספר טלפון בן 10 ספרות שמתחיל ב0. שתי רשומות עם אותו מספר זהות חייבים להיות גם עם אותו שם ושם משפחה. הכל באנגלית.

לכל לקוח יכול להיות יותר משורה אחת.

כשמפעילים את התוכנית היא מדפיסה את רשימת הלקוחות שיש להם חוב, שורה אחת לכל לקוח עם סיכום החוב, ממוינת לפי גודל החוב (הגבוהה ביותר הכי למטה), בודקת את תקינות הרשימה (כל השדות קיימים ובפורמט הנכון), אם יש טעות – מציינת אותה בהדפסה. התוכנית כמובן צריכה לסכם את כל החובות של הלקוח ולהדפיס את התאריך המוקדם ביותר של החוב שנשאר (התאריך הישן ביותר).

אחרי ההדפסה (ובלי לצאת מהתוכנית) היא מציגה PROMPT (לדוגמה: ←), עכשיו משתמש יכול לשאול שאילתות, השאילתה מתחילה במילה select, לדוגמה:

select  first name=Moshe או select dept>20.5, select date < 2/3/2022

 ניתן לשאול לפי כל פרמטר בטבלת לקוחות. התוכנית תדפיס את רשימת הלקוחות שמתאימים לשאילתה ממוינת לפי גודל החוב. אם אין אף לקוח שמתאים לערכי השאילתה, התוכנית תדפיס הודעה שאין לקוח כזה. אם אחד השדות בשאילתה נכתב בפורמט לא נכון – התוכנית תדפיס הודעת שגיאה עם ציון ברור מה לא נכון.

משתמש יכול גם להוסיף שורות עבור לקוחות קיימים או ליצור לקוחות חדשים, הוא רושם set ואז ימלא את כל השדות. צריך לבדוק אם כל נתונים נכונים, בפורמט הנכון ואם לא – להדפיס מה לא נכון. המשתמש יכול לכתוב מספר שלילי כחוב, במקרה זה – החוב של הלקוח יקטן, לדוגמה:

set first name=Moshe, second name=Berdichevsky, id=123456789, phone=0544123456, date=3/4/2022, dept=300 חוב חדש

set first name=Moshe, second name=Berdichevsky, id=123456789, phone=0544123456, date=3/4/2022, dept=-200 תשלום חוב

במידה ויוכנס יותר מידי או פחות מידי פרטים או תז של אדם עם פרטים מזהים שונים ממה שכבר נשמר במערכת אז התוכנית תעיר על כך. (לדוגמא תז: 123456789 עם שם פרטים Moshe ושם משפחה Berdich)

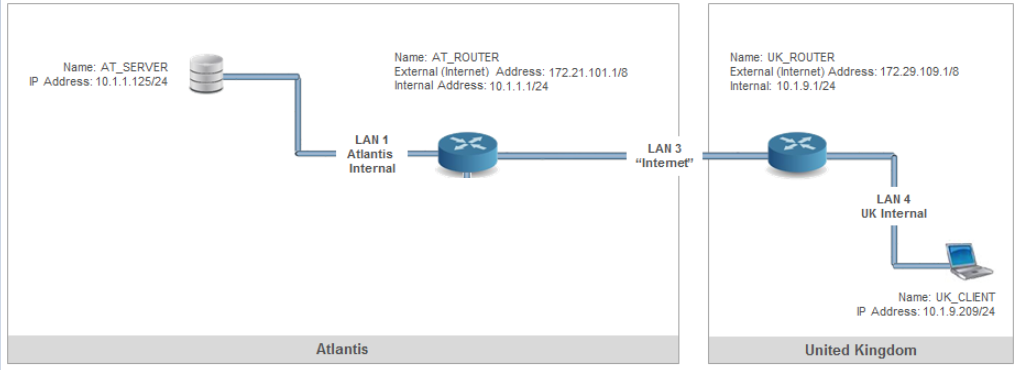
המשתמש יכול לכתוב פקודה print, התוכנית תדפיס על המסך את רשימת הלקוחות כפי שהוסבר קודם.

התוכנית תמשיך עד שהמשתמש יכתוב quit. התוכנית צריכה להיות יעילה, לא להשתמש בספריה לצורך המיון, לכתוב בעצמך.

### תחומי הידע הנדרשים בפרק

1. ידע בסיסי בC
2. מבני זיכרון (structures)
3. ניהול זיכרון
4. עבודה עם קבצים
5. רשימות מקושרות
6. מיון
7. חיפוש
8. פרסור יעיל של מחרוזות
9. מצביעים
10. מערכים
11. ספריית מתמטיקה

## חלק שני

****

הנבחן מתקין רשת לעיל.

בחלק זה נהפוך את התוכנית מחלק ראשון לשרת WEB ונכתוב client עבור ממשק משתמש. צד הclient רץ על UK\_CLIENT, צד השרת רץ על AT\_SERVET. יכולים להיות מספק קליינטים שעובדים במקביל.

### התנהגות צד הקליינט

1. קליינט מקבל את כתובת IP של השרת כפרמטר של התוכנית
2. כל קליינט פותח חמשה sockets לשרת. משתמש יכול לכתוב חמש שאילתות ואז לכתוב פקודה: send השאילתות ימוספרו לפי הסדר ויישלחו לשרת, כל שאילתה תשלח על socket נפרד.
3. בקבלת תשובה מהשרת, הקליינט ידפיס את התשובות על המסך לפי סדר של השאילתות
4. לסגור sockets

### התנהגות צד השרת

1. השרת יקבל את שם קובץ הנתונים כפרמטר של התוכנית
2. השרת יאזין עלהsocket הראשי
3. בקבלת השאילתה מקליינט - השרת ישלח אתן לביצוע לפי הסדר, עבור כל שאילתה ייפתח thread נפרד. (חשוב לשמור על הסדר כדי שהנתונים יישארו עקביים)
4. אם יש טעות בשאילתה – השרת מחזיר טעות וקליינט מדפיס אותה למשתמש
5. לסגור sockets
6. התוכנית תעבוד ללא שימוש בlock
7. השרת רץ כל הזמן ומחקה לפקודות נוספות גם אם הקליינט ייצא

### תחומי הידע הנדרשים

1. עבודה עם socket (select, accept, bind, recv, send, listen, socket, gethostbyname,getservbyname, getprotobyname**,** connect, network order, ) עם non-blocking
2. הכרות עם IPv4 והגדרות רשת בסיסיות
3. Multi-threading
4. Client-server programming
5. TCP IPC