SUMMER COURSE WEEK 2

|  |  |
| --- | --- |
|  | due to 11/7/25 @ 10:00 am Good luck! :) |

# משימה ראשונה

עליכם לכתוב תוכנית שפועלת כמחשבון כיס אינטראקטיבי לביצוע פעולות על קבוצות.

תזכורת: קבוצה (set) היא אוסף של איברים בני מניה, ללא חזרות (כלומר לכל איבר בקבוצה יש ערך שונה מכל שאר האיברים האחרים).

משימות התוכנית:

התוכנית תקבל פקודות מהקלט הסטנדרטי, היא תמשיך לקבל פקודות תפענח ותבצע אותן עד פקודת היציאה.

עליכם להגדיר, תוך שימוש בtypedef את הטיפוס set בקובץ set.h אשר מחזיק קבוצה של מספרים שלמים מהתחום הסגור 0...127 על הטיפוס להיות חסכוני מבחינת זיכרון כך למשל מערך של 128 בתים אינו חסכוני בכלל. רמז, אפשר להסתפק בסיבית אחת לכל איבר.

בנוסף עליכם להגדיר בתוכנית הראשית בקובץ myset.c שישה משתנים מטיפוס set בשמות הבאים:

SETA, SETB, SETC, SETD, SETE, SETF שבתחילת ריצת התוכנית יהיו מאותחלים עם קבוצה ריקה.

כעת, עליכם לבצע פעולות מגוונות על קבוצות. כל פעולה תופעל באמצעות פקודה שמועברת מהמשתמש בקלט לתוכנית.

מבנה הפקודות (סדר הקריאה הוא משמאל לימין):

1. **הצבת איברים בקבוצה**

רשימת ערכים מופקדים זה מזה ב-פסיקים ,שם קבוצה read\_set

הפקודה מכניסה את הערכים שברשימה לתוך הקבוצה ששמה ניתן בפקודה, רשימת הערכים אינה סדורה ומותר לערך להופיע בה יותר מפעם אחת. אך מופעים חוזרים פשוט לא יילקחו בחשבון. סוף רשימת הערכים יהיה מסומן על ידי המספר השלילי -1.

לדוגמא הפקודה:-1, 23, 98, 23, 23, 44, 76, 5, 6, 5, SETA read\_set

תציב במשתנה SETA את הקבוצה: 98, 76, 44, 23, 6, 5

לתשומת לב ההצבה יוצרת קבוצה חדשה שמחליפה את התוכן הקודם של המשתנה אם הרשימה אינה מכילה ערכים מלבד 1- שהוא ערך סימון סיום רשימה הקבוצה שתיווצר היא ריקה.

1. **הדפסת קבוצה**

שם קבוצה print\_set

הפקודה מדפיסה את איברי הקבוצה ששמה ניתן בפקודה, בסדר עולה של הערכים ולכל היותר 16 ערכים בכל שורת פלט. יש להקפיד על פורמט נאה של הדפסה. אם הקבוצה ריקה יש להדפיס הודעה: this set is empty

1. **איחוד של קבוצות**

שם קבוצה ג' ,שם קבוצה ב' ,שם קבוצה א' union\_set

הפקודה מבצעת איחוד של קבוצה א' אם קבוצה ב' ומאחסנת את התוצאה בקבוצה ג'.

תזכורת: תוצאת האיחוד היא קבוצת כל האיברים הנמצאים בקבוצה א ו/או בקבוצה ב'.

1. **חיתוך של קבוצות**

שם קבוצה ג' ,שם קבוצה ב' ,שם קבוצה א' intersect\_set

הפקודה מבצעת איחוד של קבוצה א' אם קבוצה ב' ומאחסנת את התוצאה בקבוצה ג'.

תזכורת: תוצאת האיחוד היא קבוצת כל האיברים הנמצאים בקבוצה א' וגם בקבוצה ב'.

1. **חיסור של קבוצות**

שם קבוצה ג' ,שם קבוצה ב' ,שם קבוצה א' sub\_set

הפקודה מבצעת חיסור של קבוצה א' אם קבוצה ב' ומאחסנת את התוצאה בקבוצה ג'.

תזכורת: תוצאת החיסור היא קבוצת כל האיברים הנמצאים בקבוצה א' ולא נמצאים בקבוצה ב'.

1. **הפרש סימטרי של קבוצות**

שם קבוצה ג' ,שם קבוצה ב' ,שם קבוצה א' symdiff\_set

הפקודה מחשבת הפרש סימטרי של קבוצה א' אם קבוצה ב' ומאחסנת את התוצאה בקבוצה ג'.

תזכורת: תוצאת ההפרש הסימטרי היא קבוצת כל האיברים הנמצאים בקבוצה א' או בקבוצה ב' אבל לא נמצאים בחיתוך של קבוצה א' עם קבוצה ב'.

1. **עצירת התוכנית**

stop

זוהי פקודה ללא אופרנדים, הגורמת לסיום מידי של התוכנית.

לתשומת לב: אותו שם של קבוצה יכול לשמש ביותר מאופרנד אחד באותה הפקודה! כלומר מימוש הפעולות על קבוצות צריך להתחשב באפשרות זו ולא לדרוס נתונים תוך כדי חישוב לדוגמא הפעולות שלהלן תקינות ומוגדרות היטב:

union\_set SETC, SETD, SETD

intersect\_set SETA, SETF, SETA

sub\_set SETD, SETD, SETD (what is the effect of this method?)

union\_set SETC, SETC, SETA (what is the effect of this method?)

המבנה התחבירי של הקלט:

* כל פקודה תופיע בשלמותה בשורת קלט יחידה, כלל כל האופרנדים. מותרות גם שורות ריקות (המכילות תווים לבנים space/tab)
* שם הפקודה מופרד מהאופרנד הראשון באמצעות תווים לבנים one or more space/tab
* בין כל שני אופרנדים של הפעולה יש פסיק אחד לפני ואחרי הפסיק יכולים להיות תווים לבנים space/tab בכמות בלתי מוגבלת. אסור שיהיה פסיק אחרי האופרנד האחרון
* בתחילת השורה יכולים להיות תווים לבנים space/tab בכמות בלתי מוגבלת.
* אסור שיהיו תווים מיותרים זבל בסוף השורה למעט תווים לבנים space/tab בכמות בלתי מוגבלת.
* שמות הפקודה יופיעו באותיות קטנות בלבד, ושמות הקבוצה באותיות גדולות בלבד.

אופן פעולת התוכנית:

יש ליצור ממשק משתמש ידידותי, כך שהמשתמש יוכל להבין בכל שלב של התוכנית מה עליו לעשות. בפרט, על התוכנית להדפיס הודעה או סימן (prompt) בכל פעם שהיא מוכנה לקלוט את הפקודה הבאה. התוכנית תמשיך לקלוט ולבצע פקודה אחרי פקודה. עד שתתקבל הפקודה stop.

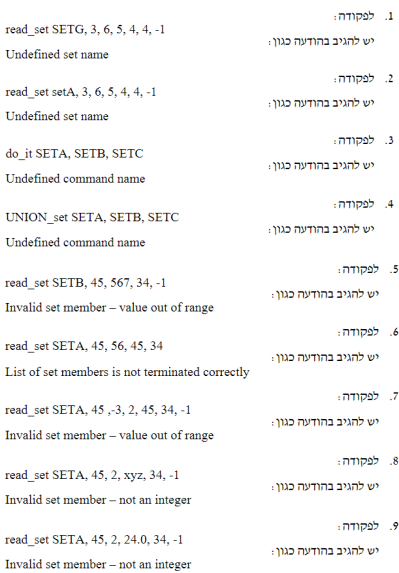
התוכנית **אינה מניחה כי הקלט תקין.** על התוכנית לנתח כח םקטדה ולוודא שאין בה שגיאות (ראו דוגמאות בהמשך). במידה ונתגלתה שגיאה, התוכנית תדפיס הודעת שגיאה פרטנית ותעבור לפקודה הבאה. בלי לבצע את הפקודה השגוייה. אין לעצור את התוכנית עם גילוי השגיאה הראשונה, אין צורך לדווח על יותר מטעות אחת בכל שורת קלט.

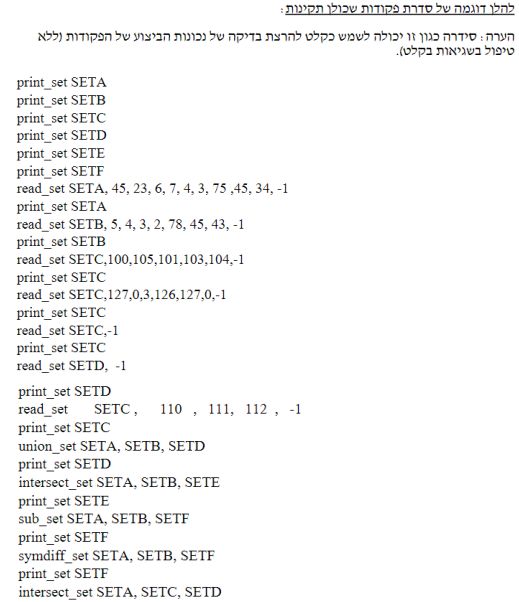
יש לטפל במצב של EOF, עצירת התוכנית של על ידי הפקודה stop מפורשת בקלט, אינה נחשבת תקינה. (גם לא בredirection). ויש להדפיס הודעת שגיאה על כך.

שימו לב: השורה האחרונה בקובץ אינה חייבת להיות ‘\n’.

להלן דוגמאות של קלט שגוי:

תמונה שמכילה טקסט, צילום מסך, גופן, מסמך

תוכן שנוצר על-ידי בינה מלאכותית עשוי להיות שגוי.



תמונה שמכילה טקסט, גופן, צילום מסך, מכתב

תוכן שנוצר על-ידי בינה מלאכותית עשוי להיות שגוי.

# משימה שנייה

קראו את פרקים 2,3,5,6,7 בספר (bgnet.pdf) שמצורף בתיקייה.

עכשיו אחרי שקראתם את הספר בפרקים שסימנו תתבקשו לבנות מערכת צ'אט אונליין עם משתמשים באופן אינטראקטיבי.

תוכניות ראשיות בקבצים server.c ו-client.c כל אחד בתת תיקייה מסודרת באופן מודולרי בתיקיית התשובות השבועית.

עליכם לכתוב תוכנת צ'אט אשר תכיל שני תוכניות:

* שרת: ינהל את כל הבקשות בצ'אט
* לקוח: יוכל להתחבר לשרת ולהיכנס לצ'אט עם שאר הלקוחות

תוכנת השרת

השרת יהיה בנוי בצורה אסינכרונית תוך שימוש בthreads ויספק את הדרישות הבאות

1. השרת יאזין לבקשות התחברות מלקוחות ויקבל עד 50 לקוחות. אם לקוח ינסה להתחבר אך אין לו מקום בשרת, השרת יפתח איתו חיבור, ישלח לו בתשובה " currently server full”ויסגור מיד את החיבור אל מול הלקוח.
2. לכל לקוח שיתחבר ייפתח thread אשר ינהל את כל התקשורת המיועדת ספציפית ללקוח הזה.
3. השרת יכיל 5 חדרים שונים לצ'אט .
4. להלן תיאור הבקשות אשר השרת יתמוך בהן לבניית הלוגיקה של השרת:
5. מוזמנים להוסיף בקשות כפי רצונכם.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **מס' בקשה** | **היררכית גישה** | **ארגומנטים** | **בקשה מהלקוח** | **פעולת השרת** |
| 100 | כל לקוח | שם וסיסמא רצויה | בקשת רישום | ירשום את הלקוח בקובץ users |
| 101 | כל לקוח | שם וסיסמא | בקשת התחברות | יחפש את הלקוח בקובץ הטקסט users, אם מצא יעביר את הלקוח להיררכית "לקוח מחובר" אם לא יחזיר כישלון. |
| 102 | לקוח מחובר | --- | בקשת רשימת החדרים בצ'אט | יחזיר רשימה של זוגות ערכים, מספר חדר ומספר לקוחות בחדר הזה. |
| 103 | לקוח מחובר | שם, מספר חדר | בקשת הצטרפות לחדר X | יעביר את הלקוח להיררכית גישה "לקוח מצורף לחדר X" וישלח הודעת שרת בחדר שלקוח (שם) התחבר. |
| 104 | לקוח מצורף לחדר X | שם, הודעה | בקשת שליחת הודעה בחדר | השרת יפיץ את ההודעה מהקלוח לכל חברי החדר. |
| 105 | לקוח מצורף לחדר X | שם | בקשה ליציאה מחדר מסוים | יינתק את הלקוח וישלח הודעת שרת בחדר שלקוח (שם) התנתק. |

תוכנת הלקוח

הלקוח יהיה בנוי בצורה אסינכרונית, בעלת שני threads:

אחד אחראי, לאחר החיבור לשרת, על האזנה להודעות מהשרת וטיפול מתאים בכל אחת מהן.

אחד אחראי על תפעול הUI, יציג הודעות בהתאם וימתין להחלטת הלקוח.

1. בהיררכיה הראשונה יאפשר הUI לבחור בין הירשמות להתחברות
2. בהיררכיה השנייה יאפשר הUI יבקש אוטומטית את רשימת החדרים, וימתין להחלטת הלקוח לאיזה חדר להתחבר. (בונוס: ניתן להוסיף יכולת לעדכן את הרשימה בלייב, מומלץ לעשות זאת באמצעות observer)
3. בהיררכיה השלישית הUI יציג כל הודעה שנשלחת מהserver לחדר, ובמקביל יאפשר ללקוח להקליד הודעה לשליחה, בעת לחיצה על enter הUI יבדוק את תוכן ההודעה, אם תוכן ההודעה הינו "`~" יפרש זאת הUI כבקשת התנתקות. אחרת, כבקשה לשליחת הודעה
4. להלן תיאור התשובות האפשריות אשר הלקוח יתמוך בהן לבנית הלוגיקה של הלקוח.

|  |  |  |  |
| --- | --- | --- | --- |
| **מס' בקשה** | **ארגומנטים** | **בקשה מהלקוח** | **פעולת השרת** |
| 200 | סטטוס הצלחה: 0: הצלחה 1-: כישלון | הצלחת רישום | הצגת הודעת רישום הצליח/נכשל וחזרה למסך אפשרויות ראשי. |
| 201 | סטטוס הצלחה: 0: הצלחה 1-: כישלון | הצלחת התחברות | הצגת הודעה פרטי התחברות נכונים/שגויים ומעבר למסך היררכיה שנייה במידה והצליח. |
| 202 | רשימת החדרים או 1- לכישלון | רשימת החדרים בצ'אט | יציג את רשימת החדרים ללקוח בצורה נאה ויאפשר לו לבחור בחדר. |
| 203 | סטטוס הצלחה: 0: הצלחה 1-: כישלון | הצלחת התחברות לחדר | יציג ללקוח את מסך הצ'אט ויאפשר לו לשלוח הודעות בחדר. |
| 204 | שם, הודעה | שליחת הודעת לקוח בחדר | יציג את ההודעה מלקוח (שם) בחדר על המסך. |
| 205 | הודעה | שליחת הודעת שרת בחדר | יציג את ההודעה מהשרת בחדר על המסך. |
| 211 | סטטוס הצלחה: 0: הצלחה 1-: כישלון | הצלחת ניתוק מחדר מסוים | יעביר את הלקוח בחזרה למסך החדרים |

\*\*\* יש לאכוף היררכיות גישה ולהגן מביצוע פעולה לא מורשית, ניתן להוסיף עוד הודעות לפי הצורך