

מטלת מנחה (ממ"ן) 22

הקורס: 20465 - מעבדה בתכנות מערכות

חומר הלימוד למטלה: פרקים 4,5,6

מספר השאלות: 1 משקל המטלה: 8 נקודות (רשות)

סמסטר: 2023'א מועד אחרון להגשה: 08.01.2023

קיימות שתי חלופות להגשת מטלות:

- שליחת מטלות באמצעות מערכת המטלות המקוונת באתר הבית של הקורס
 - שליחת מטלות באמצעות דואר אלקטרוני, באישור המנחה בלבד
- הסבר מפורט ב"נוהל הגשת מטלות מנחה"

יש לקמפל עם דגלים מקסימליים, לקבלת כל האזהרות: `-Wall -ansi -pedantic`. יש להגיש את קבצי המקור (`.c`, `.h`), קבצי ההרצה (את קבצי `.o` אין צורך לצרף), קבצי הסביבה המתאימים (כולל קבצי `makefile`), וכן קבצי קלט ותדפיסי מסך או קבצי פלט (לפי הנחיות במטלה/במפגש/באתר). קבצי התוכנית יהיו בתיקה. נדרש ששם התיקה ושם הקובץ לריצה יהיו כשם הקובץ המכיל את הפונקציה `main`, ללא הסיומת `.c`. יש להגיש תכנית מלאה (בין השאר מכילה `main`), הניתנת להידור והרצה, ומאפשרות בדיקה של כל תוצאות הריצה המגוונות ללא צורך בשינויים כלשהם בקוד המקור של התוכנית. את המטלה יש להגיש בקובץ `.zip`. לאחר ההגשה יש להוריד את המטלה משרת האו"פ למחשב האישי, ולבדוק שהקבצים אכן הוגשו באופן תקין.

שאלה 1 (תכנית ראשית בקובץ `myset.c`, ובנוסף הקבצים `set.c`, `set.h`)

עליכם לכתוב תכנית שפועלת כ"מחשב כיס" אינטראקטיבי לביצוע פעולות על קבוצות.

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

משימות התכנית:

עליכם לכתוב תכנית מחשב הקוראת פקודות מהקלט הסטנדרטי, מפענחת ומבצעת אותן. הפקודות עוסקות בפעולות על קבוצות.

עליכם להגדיר, תוך שימוש ב- `typedef`, את הטיפוס `set`, אשר מחזיק קבוצה של מספרים שלמים מהתחום הסגור `[0...127]`. על הטיפוס (מבנה הנתונים) `set` להיות חסכוני מבחינת כמות הזיכרון הנדרשת. כך למשל, מערך של 128 בתים אינו חסכוני. רמז: אפשר להסתפק בסיבית (`bit`) אחת לכל איבר בקבוצה.

בנוסף עליכם להגדיר, בתכנית הראשית, שישה משתנים מטיפוס set, בשמות הבאים :
SETA, SETB, SETC, SETD, SETE, SETF.
בתחילת ריצת התכנית, יש לאתחל כל אחד מהמשתנים לקבוצה הריקה.

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

בפקודות שיפורטו להלן, אופרנד שהוא שם של קבוצה, יהיה אחד מששת המשתנים שהוגדרו לעיל.

מבנה הפקודות המשמשות כקלט לתכנית :

לתשומת לב : סדר קריאת השדות בפקודה הוא משמאל לימין.

1.הצבת איברים בקבוצה

רשימת-ערכים-מופרדים-זה-מזה-בפסיקים, 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 (מופעים חוזרים לא יילקחו בחשבון). סוף רשימת הערכים מסומן על ידי המספר השלילי -1.

לדוגמה, הפקודה : read_set SETA, 5, 6, 5, 76, 44, 23, 23, 98, 23, -1

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

לתשומת לב : ההצבה יוצרת קבוצה חדשה, שמחליפה את התוכן הקודם של המשתנה. אם הרשימה אינה מכילה אף ערך (מלבד -1 המסיים), הקבוצה שתיווצר היא ריקה.

2.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 ההדפסה.
אם הקבוצה ריקה, יש להדפיס "The set is empty".

3.איחוד של קבוצות

שם-קבוצה-ג', שם-קבוצה-ב', שם-קבוצה-א' union_set

הפקודה מבצעת איחוד של קבוצה א' עם קבוצה ב', ואת התוצאה מאחסנת בקבוצה ג'.
תזכורת : תוצאת האיחוד היא קבוצת כל האיברים הנמצאים בקבוצה א' ו/או בקבוצה ב'.

4. חיתוך של קבוצות

שם-קבוצה-ג', שם-קבוצה-ב', שם-קבוצה-א' intersect_set

הפקודה מבצעת חיתוך של קבוצה א' עם קבוצה ב' ואת התוצאה מאחסנת בקבוצה ג'.
תזכורת : תוצאת החיתוך היא קבוצת כל האיברים הנמצאים גם בקבוצה א' וגם בקבוצה ב'.

5. חיסור של קבוצות

שם-קבוצה-ג', שם-קבוצה-ב', שם-קבוצה-א' sub_set

הפקודה מבצעת חיסור של קבוצה ב' מקבוצה א', ואת התוצאה מאחסנת בקבוצה ג'.
תזכורת : תוצאת החיסור היא קבוצת כל האיברים הנמצאים בקבוצה א' ולא נמצאים בקבוצה ב'.

6. הפרש סימטרי של קבוצות

שם-קבוצה-ג', שם-קבוצה-ב', שם-קבוצה-א' symdiff_set

הפקודה מחשבת הפרש סימטרי של קבוצה א' וקבוצה ב', ואת התוצאה מאחסנת בקבוצה ג'.
תזכורת: הפרש סימטרי הוא קבוצת כל האיברים הנמצאים בקבוצה א' או בקבוצה ב', אבל לא נמצאים בחיתוך של קבוצה א' עם קבוצה ב'.

7. עצירת התכנית

stop

זוהי פקודה ללא אופרנדים, הגורמת לסיום מידי של התכנית.

לתשומת לב: אותו שם קבוצה יכול לשמש ביותר מאופרנד אחד באותה הפקודה. מימוש הפעולות על קבוצות צריך להתחשב באפשרות זו (לא לדרוס נתונים תוך כדי חישוב).
לדוגמה, הפעולות שלהלן תקינות ומוגדרות היטב:

union_set SETC, SETD, SETD

intersect_set SETA, SETF, SETA

sub_set SETC, SETC, SETC (מה האפקט של פעולה זו?)

union_set SETA, SETA, SETF (מה האפקט של פעולה זו?)

המבנה התחבירי של הקלט:

- כל פקודה תופיע בשלמותה בשורת קלט יחידה, כולל כל האופרנדים. מותרות גם שורות ריקות (שורות המכילות רק תווים לבנים).
- שם הפקודה מופרד מהאופרנד הראשון באמצעות רווחים ו/או טאבים (אחד או יותר).
- בין כל שני אופרנדים של הפעולה יש פסיק אחד. לפני ואחרי הפסיק יכולים להיות רווחים ו/או טאבים בכמות בלתי מוגבלת. אסור שיהיה פסיק אחרי האופרנד האחרון.
- יכולים להיות רווחים ו/או טאבים בכמות בלתי מוגבלת בתחילת השורה (לפני שם הפקודה), וגם בסוף השורה (אחרי האופרנד האחרון).
- אסור שיהיו תווים מיותרים (תווי זבל) בסוף השורה, למעט תווים לבנים.
- שמות הפקודות יופיעו באותיות קטנות בלבד, ושמות הקבוצות באותיות גדולות בלבד.

אופן פעולת התכנית:

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

התכנית אינה מניחה שהקלט תקין. על התכנית לנתח כל פקודה ולוודא שאין בה שגיאות (ראו דוגמאות בהמשך). במידה ונתגלתה שגיאה, התכנית תדפיס הודעת שגיאה פרטנית, ותעבור לפקודה הבאה, בלי לבצע את הפקודה השגויה. אין לעצור את התכנית עם גילוי השגיאה הראשונה. אין צורך לדווח על יותר משגיאה אחת בכל שורת קלט.

יש לטפל גם במצב של EOF (גמר הקלט). עצירת התכנית שלא באמצעות פקודת stop מפורשת בקלט אינה נחשבת תקינה (גם לא כאשר הקלט מגיע מקובץ באמצעות redirection), ויש להדפיס הודעת שגיאה על כך ורק אז לעצור.

שימו לב: השורה האחרונה בקובץ קלט אינה חייבת להסתיים בתו 'n'.

להלן דוגמאות של קלט שגוי:

שימו לב: ייתכנו סוגים נוספים של שגיאות בקלט.
עליכם לחשוב על כל מגוון השגיאות האפשריות, ולטפל בכלן.

1. לפקודה :
read_set SETG, 3, 6, 5, 4, 4, -1
Undefined set name
יש להגיב בהודעה כגון :
2. לפקודה :
read_set setA, 3, 6, 5, 4, 4, -1
Undefined set name
יש להגיב בהודעה כגון :
3. לפקודה :
do_it SETA, SETB, SETC
Undefined command name
יש להגיב בהודעה כגון :
4. לפקודה :
UNION_set SETA, SETB, SETC
Undefined command name
יש להגיב בהודעה כגון :
5. לפקודה :
read_set SETB, 45, 567, 34, -1
Invalid set member – value out of range
יש להגיב בהודעה כגון :
6. לפקודה :
read_set SETA, 45, 56, 45, 34
List of set members is not terminated correctly
יש להגיב בהודעה כגון :
7. לפקודה :
read_set SETA, 45, -3, 2, 45, 34, -1
Invalid set member – value out of range
יש להגיב בהודעה כגון :
8. לפקודה :
read_set SETA, 45, 2, xyz, 34, -1
Invalid set member – not an integer
יש להגיב בהודעה כגון :
9. לפקודה :
read_set SETA, 45, 2, 24.0, 34, -1
Invalid set member – not an integer
יש להגיב בהודעה כגון :
10. לפקודה :
union_set SETC, SETA
Missing parameter
יש להגיב בהודעה כגון :
11. לפקודה :
union_set SETC, SETA, SETB,
Extraneous text after end of command
יש להגיב בהודעה כגון :
12. לפקודה :
print_set SETC, SETD
Extraneous text after end of command
יש להגיב בהודעה כגון :

13. לפקודה :

sub_set SETF, , SETD, SETA

יש להגיב בהודעה כגון :

Multiple consecutive commas

14. לפקודה :

intersect_set SETF SETD SETA

יש להגיב בהודעה כגון :

Missing comma

15. לפקודה :

syndiff_set, SETF, SETB, SETA

יש להגיב בהודעה כגון :

Illegal comma

להלן דוגמה של סדרת פקודות שכולן תקינות :

הערה : סידרה כגון זו יכולה לשמש כקלט להרצת בדיקה של נכונות הביצוע של הפקודות (ללא טיפול בשגיאות בקלט).

```
print_set SETA
print_set SETB
print_set SETC
print_set SETD
print_set SETE
print_set SETF
read_set SETA, 45, 23, 6, 7, 4, 3, 75, 45, 34, -1
print_set SETA
read_set SETB, 5, 4, 3, 2, 78, 45, 43, -1
print_set SETB
read_set SETC, 100, 105, 101, 103, 104, -1
print_set SETC
read_set SETC, 127, 0, 3, 126, 127, 0, -1
print_set SETC
read_set SETC, -1
print_set SETC
read_set SETD, -1
print_set SETD
read_set SETC, 110, 111, 112, -1
print_set SETC
union_set SETA, SETB, SETD
print_set SETD
intersect_set SETA, SETB, SETE
print_set SETE
sub_set SETA, SETB, SETF
print_set SETF
syndiff_set SETA, SETB, SETF
print_set SETF
intersect_set SETA, SETC, SETD
print_set SETD
```

```

union_set SETB, SETB, SETE
print_set SETE
intersect_set SETB, SETA, SETB
print_set SETB
union_set SETA, SETC, SETC
print_set SETC
symdiff_set SETC, SETA, SETC
print_set SETC
sub_set SETC,SETC,SETC
print_set SETC
union_set  SETF , SETC , SETF
print_set SETF
stop

```

ארגון קוד התכנית :

יש לחלק את התוכנית למספר קבצי מקור : set.c , myset.c , ו- set.h.

- בקובץ set.c יש לרכז את הפעולות על קבוצות. לכל פעולה יש לממש פונקציה נפרדת, ששמה כשם הפקודה בקלט (לדוגמה : union_set , print_set , read_set , וכד'). האופרנדים של הפעולה יועברו כפרמטרים של הפונקציה, לפי מפרט הפעולה כמוגדר לעיל. לא יבוצע כל קלט/פלט בקובץ זה, ולא ניתוח תחבירי של הקלט.
 - בקובץ myset.c תהיה הפונקציה main, וכן כל פעילויות האינטראקציה עם המשתמש, ניתוח הפקודות, והדפסת הודעות השגיאה. כמו כן, בקובץ זה יוגדרו ששת המשתנים מטיפוס set.
 - בקובץ set.h תהיה הגדרת טיפוס הנתונים set, וכן ההצהרות (אב טיפוס) של הפונקציות הממומשות בקובץ set.c. יש לכלול (#include) את הקובץ set.h בקבצי המקור האחרים.
 - אפשר לבנות קבצי מקור נוספים (למשל : קובץ המכיל פונקציות עזר לניתוח הקלט, וכד').
- הקלט לתכנית הוא מ-stdin, ויכול להגיע מהמקלדת או מקובץ (באמצעות redirection בעת הפעלת התכנית). לנוחיותכם, הכינו מספר קבצי קלט והשתמשו בהם שוב ושוב כדיבוג התכנית. בכל קובץ קלט תהיה סדרה של פקודות על קבוצות.
- על התכנית להדפיס הודעת בקשה ידידותית לקלט עבור כל שורת קלט (כל פקודה). כמו כן, לפני הניתוח של שורת הקלט, על התכנית להדפיס את השורה לפלט בדיוק כפי שנקראה. זאת כדי שניתן יהיה לראות בפלט את הפקודות המקוריות, גם כאשר הקלט מגיע מקובץ.
- חובה לצרף להגשה הרצות בדיקה (אחת או יותר), המדגימות את השימוש בכל הפעולות על קבוצות ובכל ששת הקבוצות המוגדרות, וכן את הטיפול בכל מגוון השגיאות בקלט. רמז : מומלץ להכניס בקלט פקודת הדפסה של קבוצת התוצאה אחרי כל פעולה, כדי להראות שהתוצאה אכן נכונה (ראו לעיל הדוגמה של סדרת פקודות תקינות).
- יש להגיש תדפיס מסך (או קובץ פלט) של כל ההרצות. אם תשתמשו בקבצי קלט, יש להגיש גם קבצים אלה.

לתשומת לבכם : לא תינתן דחיה בהגשת הממ"ן, פרט למקרים מיוחדים כגון מילואים או אשפוז. במקרים אלו יש לבקש ולקבל אישור מראש ממנחה הקבוצה.

בהצלחה !