Here is a copy of my answers to questions asked by a MaTaM student.
I hope it will help other students (not necessarily MaTaM students).

Originally, I took screen-shots - this is why parts are duplicated.
I hope I'll have time to repair it.

Enjoy
Yechiel

טופס המבחן: https://webcourse.cs.technion.ac.il/234122/Spring2014/ho/WCFiles/14SpringB-solve4students.pdf

========== שאלות חלק 1 (C++) ==========

ניתן לשאול אותך לגבי פתרון שלך עבור מבחן במת"ם בעבר (שנת 2014, אביב, מועד ב, שאלה 3, סעיף א **- מצורפת תמונה כאן במייל**)?
יש כמה דברים שאני לא בטוח לגביהם, ולאחר שניסיתי להתייעץ לגביהם עם גורמים אחרים, קיבלתי תשובות לא ברורות, וחלק גם לא ידעו לענות על חלק מהשאלות.

1. מדוע עבור הבנאי שמקבל "קטע יחיד" לא בדאי להגדיר אותו כ-explicit?
   (ובפרט, האם זה נחשב כ"טעות"?).
2. מדוע בפתרון העדפת להשתמש במערך ולא בווקטור? לבאורה, זה נראה ששימוש בווקטור יחסוך בעבודה עם זיכרון ובפרט נוכל להשתמש בדיפולטיביים עבור ה- operator=,c'tor,d'tor.
   אז למה לבאורה לא לעשות זאת כך?
3. מדוע צריך בכל פעם כשנדרשים לממש את operator+ , לממש גם את operator+= ?
4. מדוע המתודות unite ו-intersect מחזירות את הפלט by-reference (מודגש במסגרת בצבע חום בתמונה) ? לבאורה לפי התרגיל זה נראה שצריך ליצור אוסף *חדש* , ולכן בפרט נצטרך להחזיר אותו by-value (שהרי כתוב "ליצירת אוסף **שלישי** - מכך אני מבין שצריך ליצור אוסף חדש, ולכן by-value, אחרת אם נחזיר by-reference אז מה שניצור בתוך המתודה יימחק כשנצא מהמתודה ולא נוכל בפועל להחזיר "ממש אותו").
5. לגבי ה "operator+" בהקשר של המחלקה SegCollection,
   ביצד ניתן להבין מהי הדרך הטובה להגדיר אותו?
   בלומר, האם יש להגדיר "צורה אחת שלו" כך שהוא מקבל קודם אובייקט מטיפוס SegCollection ולאחר מכן אובייקט מטיפוס Segment.
   או <u>שבנוסף</u> לאופרטור בצורה שציינתי לעיל, צריך להגדיר צורה נוספת שלו
   כך שהוא מקבל קודם אובייקט מטיפוס Segment ולאחר מכן אובייקט מטיפוס SegCollection?
   (ובכלל, כשמבקשים ממני להגדיר עבור מחלקה כלשהי את operator+, מהי הדרך הנכונה לעשות זאת?)
6. איך ניתן יהיה לממש את operator-() עבור SegCollection?
   הבעיה שלי לגביו היא העובדה שאם לוקחים קטע כלשהו, ומחסירים ממנו קטע אחר, אז תוצאת החיסור עשויה להיות *2 קטעים* , ואז לא ברור לי כיצד ניתן להחזיר דבר כזה.
   בלומר, למשל יש לי את הקטע: [1,5]
   ואני רוצה להסיר את הקטע: (2,3)
   אז אני אמור לקבל: [1,2] , [3,5]
   ולא ברור לי כיצד ניתן להחזיר 2 קטעים בצורה שתואמת לכללים שנלמדים בכיתה.

   (לכאורה, ניתן פשוט להחזיר את זה ב-vector כך שיהיה בו לכל היותר 2 "קטעים",
   האם זה סביר היה לממש אותה [במידה והיה נדרש] בצורה כזו?)

========== תשובות חלק 1 ==========

לשאלתך הראשונה: ניתן לשאול אותי, ואני אפילו עונה (כפי שאתה מבין כעת).
לשאר השאלות אענה _לא_ לפי הסדר, כי כך קל לי יותר להסביר.

3. אני מציע לקרוא את השקפים שלי בנושא העמסת פעולת + (החל מהגרסה ראשונה, שאני מציין שאינה כל-כך טובה ועד המימוש בעזרת =+ המסומן כ- Best version).
לא בדקתי כיצד אחרים מסבירים את הרעיונות, לכן אני מעדיף להפנות לשקפים שלי.
השקפים המעודכנים יותר הם מאביב 2016. השקפים של אביב 2014 מספיקים עבור נושא זה (לא שונו בהרבה).
<u>בשורה התחתונה</u>: אם יש + ואין =+ אתה מפתיע את המשתמש במחלקה (מעבר לפגיעה בביצועים)
- שים לב להערה הרלוונטית בפתרון.

4. קרא את ההערה בשורה שמעל המלבן הירוק. אין אופרטור טבעי לאיחוד וחיתוך, לכן ממומש כפונקציה
- אבל התשובה ל- (3) רלוונטית במלואה.

2. לשימוש בווקטור של STL יש יתרונות, אבל רציתי להציג פתרון שיסביר גם לאלה שאינם מכירים את STL.
אני גם מתנגד לשימוש מוגזם ב STL בקורס, במקום תכנות בסיסי:
אם משתמשים רק ב STL, איך תלמדו לבנות מחלקות דמויות STL (כולל שיפור מחלקות של STL, למשל וקטור)?
<u>הערה חשובה</u>: הטיפוס של all בפתרון הוא שגוי (טעות קולמוס, אפשר לראות לפי הקוד של הבנאי, סעיף ב').
צ"ל Segment<T> . גם בבנאי יש שגיאה כשכתוב new T [ ] , אבל החלק הנכון הוא כשכתוב all[0] = s.

1. להגדיר explicit זו אינה טעות (כתוב בהסבר "נָחוּת"), אבל אם יש פונקציה שצריכה לקבל "אוסף"
אבל אנחנו רוצים לשלוח "קטע" בודד,
מדוע צריך לשלוח במפורש בנאי עבור ה"אוסף" אם אפשר לשלוח "קטע" בודד, והמהדר ישלים את החסר
(ההבדל בין "קטע" לבין "אוסף עם קטע בודד" הוא <u>לרוב זניח</u>). explicit נוצר למנוע טעויות מסוג אחר:
למשל, אתה שולח 5 כארגומנט לפונקציה כשאינך מודע לכך שהוא הופך למערך בגודל 5.

5. כיוון שהבנאי איננו explicit זה לא משנה (כשה"קטע" נמצא <u>בצד ימין</u>) - יתרון נוסף להמנע מ- explicit.
לגבי הופעתו בצד שמאל, פונקציה מזערית (שורה אחת) שהיא inline פותרת את הבעיה (וראוי לממש זאת).

6. אפשרי בקלות יחסית, כי האוסף ממומש ע"י מערך של "קטעים". ראה הסבר בפתרון סעיף ד'.

========== שאלות חלק 2 ==========

תודה רבה על התשובה , זה שפך אור על התרגיל , מעריך את ההתייחסות.

<u>עדיין דבר שלא נותר ברור זה לגבי האיחוד והחיתוך.</u>

מדוע בפתרון ב-union_with וב- intersect_with מחזירים את הפלט ב-by-reference
(כמו שכתוב בפתרון "&SegCollection").

למה קשה לי להבין את זה?

כי בפעולות האלה (כפי שכתוב בתרגיל - "ליצירת אוסף **<u>שלישי</u>**") אנו יוצרים אוסף חדש. לכן בפרט, האוסף של התוצאה (אוסף האיחוד או אוסף החיתוך) יהיה "משתנה זמני" - ולא ניתן להחזיר משתנה זמני by-reference, אלא רק by-value. לכן איך בכל זאת זה מוחזר שם by-reference?

========== תשובות חלק 2 ==========

אבל בפתרון כתוב שהפונקציות עם with בִּשְׁמָן מתפקדות כמו =+ (כלומר, משנות את this) ולכן הן מחזירות את this כרפרנס,
והפעולה union תמומש בעזרת union_with כמו שהמימוש של + משתמש ב =+.

========== שאלות חלק 3 ==========

כד"א, אם למשל אני רוצה לממש את האופרטור חיסור (זה שמוריד "קטעים" מהמבנה) עבור SegCollection,
אז הוא היה צריך לעשות שימוש באופרטור חיסור שצריך להגדיר עבור Segment כך שהוא (האופרטור חיסור של Segment) יחזיר מערך שגודלו 2 לכל היותר (או וקטור שגודלו בסופו של דבר יהיה 2 לכל היותר) של קטעים ?

[אני חושב שה-operator חיסור של Segment אמור להחזיר וקטור שלכל היותר גודלו בסופו של דבר יהיה 2, כי כאשר מורידים מקטע בלשהו איזשהו קטע, אז עשוי להיות "פיצול" ואז התוצאה של החיסור תהיה למעשה 2 קטעים - מה שהיה באותו "קטע" במיקום של לפני הקטע שהסרנו, ומה שהיה לאחריו]

זו הסתבלות נכונה או שאני מפספס כאן משהו?

========== תשובות חלק 3 ==========

בעיקרון זה נכון; כלומר, הפתרון ה"אידאלי" בעולם חסר מגבלות הוא
שחיסור קטעים יחזיר "אוסף" קטעים (שבו לכל-היותר שני קטעים).

אבל יכולות להיות כמה הסתייגויות:
0. בעולם שבו אין "אוסף קטעים", זה לא הגיוני לחייב קיום עצמים מסוג חדש
(משך אלפי שנים הסתדרנו עם "אין שרש למספר שלילי").
1. אפשר להחליט (כן, זו החלטה) שאם התוצאה היא שני קטעים, אז תוצאת החיסור אינה מוגדרת
(למשל 3 לחלק ל- 2 אינו מוגדר בשלמים) - זה מקרה פרטי של 0.
2. אפשר להחליט (כן, ...) שחיסור קטעים מחזיר מערך של שני קטעים (שהאיבר השני יכול להיות 0)
(למשל, 3 לחלק ל- 2 בשלמים מחזיר 1 ושארית 1)
- זו דרך לעקוף את הבעיה של סעיף 0.

הפתרון (בחיי המעשה), עבור הבעיה שבמבחן, תלוי במצב:
1. אם יש לך גישה אל (ורשות לשנות את) המחלקה קטע, בחר בפתרון 2, ע"י הוספת פעולה למחלקה
(עדיף "מוסרית" - כדי לא לחייב את משתמשי "קטע" להשתמש ב"אוסף" - זה יוצר תלות מעגלית בין המחלקות).
2. אם אין לך רשות לשנות את המחלקה "קטע", או שאינך רוצה, אתה יכול להגדיר את הפעולה כ"פעולה חיצונית",
כי כל הנתונים נגישים. כתוצאה מכך, אם תרצה, תוכל לצרף את פעולת החיסור למחלקה "אוסף"
- כפעולה חיצונית - ואז כבר תוכל להגדיר את התוצאה כ"אוסף".
3. אפשר לשלב את השניים למעלה ברמות שונות.

הערה: גם במקרה (1) הפעולה תהיה חיצונית, כי אין משמעות לפעולה =- כשהתוצאה היא שני קטעים (דוגמה ל - בלי =-).

הדיון הזה מסביר את דעתי (לא מקורי שלי) על ההבדל בין "מתכנת" לבין "מהנדס תוכנה":
מהנדס נדרש לבחור בין פתרונות אלטרנטיביים.

========== סוף C++ ==========

========== שאלת C חלק 1 ==========

אשמח לדעת האם באמת יש כאן טעות או שאני מפספס משהו.
מצורף לינק שפרסמת בעבר מבחן במת"ם:
https://webcourse.cs.technion.ac.il/234122/Spring2014/ho/WCFiles/14SpringB-solve4students.pdf
ואני שואל לגבי שאלה 1.א .

במסגרת הגדרת ה-ADT של Expression. הגדרת מבנה (functions) שמקבץ את כל הפונקציות
שקשורות ל-Element (ובפרט גם פעולות חשבוניות).
ובנוסף, ב- ExprCreate הגדרת אותה כ-כזו שמקבלת ארגומנט מטיפוס Functions.
אבל לא היה צריך לציין גם פונקציית יצירה עבור אותו Functions?
(הרי לא ניתן באמת ליצור עצם מטיפוס Functions מבחוץ כי ה-struct שלו מוסתר בקובץ c.
אז לכאורה יש 2 אפשרויות, או להוציא את המבנה של functions לקובץ h , וזה נראה פחות כדאי לדעתי.
או, ליצור FunctionsCreate בקובץ Expressions.h - וזה לא מופיע בפתרון.
אני לא בטוח אם זה לא מופיע כי פשוט התעלמת מזה לצורך הפתרון,
או שיש משהו אחר שדרכו פותרים את הבעיה הזו).

אשמח לקבל תשובה בנושא כדי לדעת אם זה נכון או לא.

========== תשובה לשאלת C חלק 1 ==========

הסיבה העיקרית היא שגם אין פונקציות יצירה ל- expression ול- operation:
גם בניסוח השאלה וגם בהסברים של הפתרון, יש הרבה מאמץ להגדיר בדיוק מה לא צריך לממש.
יש אפילו חלקים ממומשים בפיתרון שעליהם נאמר שלא לצפות שיהיו בתשובות הסטודנטים
- מטרתם להבהיר את הפתרון למתרגלים הבודקים וגם לסטודנטים כשישוו את פתרונם לפתרון שלנו.

אבל יש סיבה נוספת:
ברור שאין פונקציית יצירה ל Number, מפני שזה באחריות המשתמש (כמו Element עבור Set).
אבל בהחלט ייתכן שיחד עם הבחירה של Number, נצטרך להחליט על operations:
למשל, כמקרה קיצוני, אם רוצים ש Number ייצג מטריצות ריבועיות מסדר N, אז יש פעולות המיוחדות להן.
לכן functions וגם enum Oper צריכים להיות בקוד של המשתמש - אפילו לא בכותרים של expression.

==========

מקווה שעזרתי
בהצלחה
יחיאל

```
Yechiel
--------------------------------------------------------------------------------
        Those who look a little bit farther see much more
```