

תכנות מתקדם – 150024

תרגיל בית מספר 8

תבניות ומיכלי STL – STL Containers, Templates

שים לב:

- א. הקפד/י על קריאות התכנית ועל עימוד (Indentation).
- ב. הקפד/י לבצע בדיוק את הנדרש בכל שאלה.
- ג. בכל אחת מהשאלות יש להגדיר כל מחלקה ב 2 קבצים נפרדים. קובץ H וקובץ CPP.
- ד. בכל אחת מהשאלות יש להגדיר פונקציות עזר במידת הצורך עבור קריאות התכנית.
- ה. יש להגיש את התרגיל על פי ההנחיות להגשת תרגילים (המופיע באתר הקורס) וביניהם:
השתמש/י בשמות משמעותיים עבור המשתנים.
יש לתעד את התכנית גם עבור פונקציות אותם הנך מגדיר/ה וכן על תנאים ולולאות וקטעי קוד מורכבים, ובנוסף, **דוגמת הרצה לכל תכנית בסוף הקובץ!**

הערה חשובה: לכל תרגיל בית מוגדר שבוע אחד בלבד להגשה, אלא אם כן קיבלת הוראה אחרת מהמרצה שלך. תיבות ההגשה הפתוחות לא מהוות היתר להגשה באיחור.

שאלה 1

שאלה זו מתבססת על התוכנית שנכתבה בתרגילים 6 ו-7. בתרגיל זה, נקלוט מופעים של סטודנטים מכל הסוגים (BA, MA, או PHD) לתוך רשימה מקושרת של STL, אח"כ נסכנים סטודנטים מכל סוג בנפרד לווקטור של STL, נמיון כל וקטור לפי הקריטריון בהתאם לסוג לסטודנטים, ונשתמש בתור ובמחסנית של STL כדי להדפיס את נתוני הסטודנטים לפי הסדר מהדירוג הגבוה לנמוך.

להלן ההוראות לביצוע משימת התרגיל.

א. יש להתאים את הקוד של התרגיל 7 שלך כדי להמשיך ממנו לתרגיל הנוכחי:

1. לשנות את הרשאת הגישה של המתודה studType במחלקה Student ובכל המחלקות המתבססות עליה מ-protected ל-public.
2. לוודא שבמחלקת הבסיס Student מוגדר הורס וירטואלי, ובמחלקה עם זיכרון דינמי (BA) ממומש "כלל החמישה" **במלואו** (המתודות של מיכלי STL ישתמשו בכולם).
3. הוסיפו במתודה print במחלקה Student את פלט *** Scholarship *** במידה והסטודנט זכאי למלגה, לפי דוגמת ההרצה.
4. הוסיפו בפונקציה הגלובלית addStudent את האפשרות לקלוט 0 לסיום הקלט (ראו את דוגמת ההרצה). במקרה זה הפונקציה תחזיר nullptr.

ב. הגדירו אופרטורים < להשוואה בין שני מופעים של סטודנטים מאותו סוג:

1. במחלקה Student הגדירו אופרטור < וירטואלי ללא מימוש עם החתימה:
`virtual bool operator < (const Student& rhs) const = 0;`

2. המימוש עבור המחלקות BA ו-MA יוגדר רק ב-BA ויבצע את הלוגיקה הבאה:

- אם הסוגים הסטודנט במופע השמאלי ובמופע הימני אינם זהים, יזרוק חריגה עם ההודעה "cannot compare students of different types"

- אם הסטודנט המיוצג במופע השמאלי אינו זכאי למלגה והסטודנט במופע הימני זכאי, יחזיר true

- אם הסטודנט במופע השמאלי זכאי למלגה ובמופע הימני לא, יחזיר false

- אם שניהם זכאים או שניהם אינם זכאים למלגה, יחזיר true אם ממוצע הציונים במופע השמאלי נמוך מהממוצע במופע הימני, אחרת יחזיר false.

3. המימוש במחלקה PHD יעבוד לפי אותה הלוגיקה שהמימוש ב-BA עם ההבדל שכאשר שני הסטודנטים שווים מבחינת הזכאות למלגה, המופע "הקטן" יהיה זה עם פחות שעות מחקר.

ג. יש לשכתב מחדש את פונקציה הראשית. ב-main החדשה יש לבצע את הפעולות הבאות:

1. לקלוט מהמשתמש מופעים של סטודנטים מכל הסוגים בעזרת הפונקציה addStudent עד שיבחר לסיים ע"י הקלט 0, ולהכניס אותם לרשימה מקושרת list של STL (מה צריך להיות הטיפוס של האיברים בהגדרת הרשימה?)

2. להגדיר שלושה מיכלים (containers) מסוג STL vector עבור סטודנטים מכל סוג בנפרד (שמו לב: האיברים בהם צריכים להיות מופעים ולא מצביעים, על מנת שהפונקציה sort בהמשך תזמן את האופרטור < שהגדרתם). אח"כ יש להכניס כל מופע מהרשימה לווקטור המתאים לפי סוגו שיוחזר מהמתודה studType (איזו המרה יש לעשות?) ולשחרר את הזיכרון של המופע ברשימה.

3. למיין כל אחד משלושת הווקטורים ע"י הפונקציה sort מהספרייה <algorithm>, המקבלת שני איטרטורים. שימו לב: המיין יהיה בסדר לא-יורד, מהקטן לגדול.

4. להגדיר שני תורים queue של STL, אחד למופעים מסוג BA והשני ל-MA, ומחסנית stack לאיברים מסוג PHD. להכניס את המופעים של BA ושל MA מהווקטורים לתורים במעבר מסוף הווקטור להתחלה, כך שהסטודנטים הטובים יהיו בראש התור (השתמשו ב-const reverse iterator), ואת המופעים של PHD הכניסו למחסנית במעבר מתחילת הווקטור לסופו, כך שהסטודנט הטוב ביותר יהיה בראש המחסנית.

5. להדפיס את כל המופעים – קודם מסוג BA, אח"כ MA ואח"כ PHD תוך כדי מחיקתם מהתור או המחסנית עד שזה מתרוקן.

כתוצאה, כל הסטודנטים שנקלטו יודפסו בשלוש רשימות, כל תואר בנפרד, מדורגים לפי הקריטריונים שהגדרנו.

שימו לב שאחרי שחרור המופעים מהרשימה הראשונה, אין צורך בשחרור זיכרון נוסף ע"י delete - כל הזיכרון ישוחרר ע"י הורסים (destructors) באופן אוטומטי.

דוגמא להרצת התכנית:

```
Enter degree: 1 for BA, 2 for MA, or 3 for PhD, 0 to finish
1
enter id, first name, last name, number of courses
123 Reuven Reuveni 10
and enter a list of student grades
100 100 100 100 100 100 100 100 100 100
Enter degree: 1 for BA, 2 for MA, or 3 for PhD, 0 to finish
1
enter id, first name, last name, number of courses
234 Shimon Shimoni 10
and enter a list of student grades
95 95 95 95 95 95 95 95 95 95
Enter degree: 1 for BA, 2 for MA, or 3 for PhD, 0 to finish
1
enter id, first name, last name, number of courses
345 Levi Levi 10
and enter a list of student grades
99 99 99 99 99 99 99 99 99 99
Enter degree: 1 for BA, 2 for MA, or 3 for PhD, 0 to finish
1
enter id, first name, last name, number of courses
444 Ploni Almoni 0
and enter a list of student grades
Enter degree: 1 for BA, 2 for MA, or 3 for PhD, 0 to finish
2
enter id, first name, last name, number of courses
555 Yehuda Yehuda 7
and enter a list of student grades
95 95 95 95 95 95 95
enter 1 if the student does research and 0 if not
1
Enter degree: 1 for BA, 2 for MA, or 3 for PhD, 0 to finish
2
enter id, first name, last name, number of courses
678 Dan Dani 7
and enter a list of student grades
97 97 97 97 97 97 97
enter 1 if the student does research and 0 if not
1
```

Enter degree: 1 for BA, 2 for MA, or 3 for PhD, 0 to finish

3

enter id, first name, last name, number of courses

789 Naphtali Naphtali 2

enter number of research hours:

26

Enter degree: 1 for BA, 2 for MA, or 3 for PhD, 0 to finish

7

Exception: no such degree

Enter degree: 1 for BA, 2 for MA, or 3 for PhD, 0 to finish

3

enter id, first name, last name, number of courses

890 Gad Gad 2

enter number of research hours:

32

Enter degree: 1 for BA, 2 for MA, or 3 for PhD, 0 to finish

3

enter id, first name, last name, number of courses

901 Asher Asher 1

enter number of research hours:

36

Enter degree: 1 for BA, 2 for MA, or 3 for PhD, 0 to finish

0

BA student *** Scholarship ***

ID: 123

Name: Reuven Reuveni

Grades: 100 100 100 100 100 100 100 100 100 100

Average: 100

BA student *** Scholarship ***

ID: 345

Name: Levi Levi

Grades: 99 99 99 99 99 99 99 99 99 99

Average: 99

BA student

ID: 234

Name: Shimon Shimoni

Grades: 95 95 95 95 95 95 95 95 95 95

Average: 95

BA student

ID: 444

Name: Ploni Almoni

Grades:

Average: 0

MA student *** Scholarship ***

ID: 678

Name: Dan Dani

Grades: 97 97 97 97 97 97 97

Average: 97

research: YES

MA student *** Scholarship ***

ID: 555

Name: Yehuda Yehuda

Grades: 95 95 95 95 95 95 95

Average: 95

research: YES

PhD student *** Scholarship ***

ID: 890

Name: Gad Gadi

Number of research hours: 32

PhD student *** Scholarship ***

ID: 789

Name: Naphtali Naphtali

Number of research hours: 26

PhD student

ID: 901

Name: Asher Asher

Number of research hours: 36

(המשך בעמוד הבא)

הצעה לייעול הרצות בדיקה: הכינו קובץ טקסט עם כל הקלט שהמשתמש אמור להקליד לבדיקה, ותעשו "העתק-הדבק" של כל המלל הזה לחלון התוכנית. אם תשימו בתוכנית נקודת עצירה (breakpoint) ותריצו את התוכנית ב-Debug, התוכנית תעצור כאשר המלל שעדיין לא נקלט ממתין בתור וייקלט כשתמשיכו את ביצוע התוכנית. ניתן לעשות שימוש חוזר בקובץ בלי להקליד הכל מחדש בכל הרצה, ולשמור קבצים למקרי בדיקה שונים.

להלן הקלט עבור דוגמת ההרצה למעלה:

```
1
123 Reuven Reuveni 10
100 100 100 100 100 100 100 100 100 100
1
234 Shimon Shimoni 10
95 95 95 95 95 95 95 95 95 95
1
345 Levi Levi 10
99 99 99 99 99 99 99 99 99 99
1
444 Ploni Almoni 0
2
555 Yehuda Yehuda 7
95 95 95 95 95 95 95
1
2
678 Dan Dani 7
97 97 97 97 97 97 97
1
3
789 Naphtali Naphtali 2
26
7
3
890 Gad Gadi 2
32
3
901 Asher Asher 1
36
0
```

בהצלחה רבה!!