



תרגיל ראשון

שפת C, תחביר C++. מערכת החריגות של C++. העמסת אופרטורים.

מטרות התרגיל

היכרות עמוקה יותר עם C++ שיפור יכולות עיצוב קוד. לימוד עצמי.

הנחיות כלליות

- אסור לתכנית שכתבתם לקרוס משום סיבה.
- אין לבצע הדפסות מיותרות למסך.
- כאשר המשתמש הכניס קלט שאינו תקין יש להודיע לו על כך.
- יש להימנע ככל הניתן משכפול קוד.
- יש להפריד את התכנית באופן חכם לקבצי הצהרה וקבצי מימוש. אין לממש פונקציות שאינן בסיסיות בקובץ הצהרה.
- במידה ואתם כותבים קוד שאינו פשוט, יש להוסיף שורת תיעוד. אין צורך לתעד קוד מובן מאליו.
- יש לכתוב פונקציות קצרות ככל האפשר.
- ניתן להניח הנחות מפשטות במידה ואין פגיעה בדרישות התרגיל.
- יש להשתדל לא לבזבז משאבי מערכת – ובפרט, לנקות זיכרון מוקצה ולסגור קבצים שנגמר השימוש בהם.
- הפעילו שיקול דעת.

הוראות הגשה

- את התרגיל יש להגיש במערכת הסבמיט
- יש להגיש את התרגיל לקבוצת תרגול 03, לא משנה לאיזו קבוצה אתם רשומים
- את קבצי הקוד יש להגיש בתיקייה בשם source. בנוסף, יש לכלול קובץ בשם details.txt המכיל שם מלא ותעודת זהות
- חובה לוודא כי המערכת קיבלה את ההגשה שלכם! תרגיל שלא אושר על ידי המערכת, דינו כלא התקבל.

חלק א - מחלקת InInt

בתרגיל זה תיצרו את מחלקת InInt. המחלקה היא מחלקה המכילה מספר שלם, חיובי או שלילי אשר אינו מוגבל בגודלו. המחלקה תתמוך בכל האופרטורים הרגילים עבור Integer, ובנוסף באופרטור הדפסה (פלט) ואופרטור קלט המקבל מחרוזת. שימו לב שיש לממש גם את אופרטורי הביטים, כדוגמת <, ~, &.

אין צורך לממש את כל אופרטורי ההשמה. מתוך אופרטורי ההשמה יש לממש את +=, >>=, &=. את האופרטור = תממשו בחלק ב'.

אין צורך לממש את אופרטורים שאינם מוגדרים על int, כדוגמת האופרטור [].

ישנם אופרטורים שיש לממש מספר פעמים – לדוגמא, האופרטור ++ מוגדר מימין ומשמאל, והאופרטור << מוגדר עבור פעולות ביטים ועבור קלט.

אין צורך לתמוך בפעולות עם מספרים רגילים, אבל חובה להכין בנאי (constructor) שמקבל Integer / long בתור פרמטר.

בנאים הנדרשים למחלקה, הינם: בנאי שלא מקבל כלום ומאתחל את המספר ל-0, בנאי המקבל מחרוזת (מערך של char) ומכין ממנה מספר, ובנאי המקבל מספר שלם רגיל.

חלק ב' – Reference counting (ביוס 10 נק')

בחלק זה תממשו Reference counting.

ניתן להרחיב על Reference counting באינטרנט.

שימו לב שעליכם להקפיד שלא תהיה דליפת זיכרון בשום שלב של התכנית.

חלק ג' – הכנה לבדיקה אוטומטית

עליכם לקרוא למחלקה InInt.

לצורך שימוש במחלקה, נעשה include לקובץ "LargeIntegers.h". זאת אומרת, שלאחר שכתבתי

```
#include "LargeIntegers.h"
```

אני יכול להשתמש במחלקה כנדרש בתרגיל.

שורת קימפול (חייב לעבוד אצל כולם):

```
g++ *.cpp *.h -std=c++1y
```

הוספתי קובץ דמו

בהצלחה!

נספח

רשימת האופרטורים שצריכים לממש:

+, -, *, /, %, ^, &, |, ~, <, >, <=, >=, ==, +=, >>=, &=

בנוסף, את האופרטורים הבאים יש לממש במספר דרכים:

1. ++, -- גם postfix וגם prefix

2. <<, >> גם דחיפת ביטים וגם קלט/פלט

וכמובן, יש לממש את אופרטור ההמרה ל-int. במידה והמספר גדול מ-int רגיל, יתבצע קיצוץ של הביטים המיותרים.

דוגמת קוד שאמור לרוץ:

```
int main()
{
    InfInt i = "1234123561269128075127893561293785612897356120893189627305";
    i = i << 56;
    InfInt j = "-235182305981623978569187236509123758039467349056546";
    cout << "i: " << i << endl << "j: " << j << endl;
    InfInt k = i * j;
    if (i++ == i) {
        cout << "??";
    }
    i = i*1000;
    bool result = (k % i >= j);
    return 0;
}
```

הנחיות כלליות לעיצוב קוד, ניתן למצוא כאן: <https://google.github.io/styleguide/cppguide.html>

ההנחיות אינן מחייבות