המחלקה להנדסת מערכות תקשורת
אוני' בן גוריון בנגב

הנחיות הגשה:

1. הגשה הינה ביחידים בלבד.
2. יש להגיש את כל קבצי העבודה במערכת ה VPL ולוודא שאין שגיאות קומפילציה במערכת.
3. כל מחלקה תכלול קובץ header וקובץ cpp.
4. בתרגיל זה אתם רשאים להשתמש בספריות iostream, cmath, vector בלבד.
5. הקפידו על הוראות ברורות וקריאות למשתמש לפני כל פעולת קלט.
6. יש להקפיד על שמות המחלקות והשיטות כפי שמצוין בעבודה (הקפידו על אותיות גדולות וקטנות).
7. הקפידו על תכנות נכון, זהו חלק מהציון:
   a. • שמות משתנים בעלי משמעות.
   b. • שימוש חוזר בקוד.
   c. • הקפדה על הזחות.
   d. • כתבו הערות שמסבירות את הקוד שלכם. אם נראה לכם שכתבתם מספיק הערות, תוסיפו עוד כמה.
8. אין חובה להדפיס הדפסות מפונקציות destructor/constructor/set/get.
9. destructor מופעל אוטומטית בסוף ה scope של האובייקט ואין צורך לקרוא לו.
10. יש לשחרר את כל הזיכרונות שהוקצו דינמית ביציאה מכל פונקציה ובסיום התוכנית.
11. בראש כל קובץ יש להוסיף בהערה את הטקסט הבא:

```
*/Assignment C++: 1
  Author: Israel Israeli, ID: 01234567
/*
```

כמובן שעליכם להחליף את הפרטים בפרטים שלכם.

12. נסו להבין מה רוצים לבדוק בתרגיל. לכל דף עבודה ישנה כותרת ובה מפורטים נושאי התרגילים לדף העבודה. אפשר להסיק ממנה מה מצופה בתרגילים עצמם. יש להפעיל שיקול דעת במקרה שיש סעיף שאינכם בטוחים מה לעשות בו. אם עדיין לא ברור, העלו שאלה בפורום הרלוונטי לתרגיל. נא לא לשלוח שאלות שאתם חושבים שהן שאלות שניתן לפתור בעזרת הגיון בריא.
13. הארכות יינתנו רק במקרי מילואים, אבל ומחלה חריפה (שלא נדע) ובצירוף אישורים מתאימים.

תרגיל מס' 2

נושא התרגיל: העמסת אופרטורים ושימוש בווקטורים

<u>הערה חשובה</u>: רשימת המתודות היא **<u>חלקית</u>** בלבד. היא מהווה את המינימום הנדרש ממכם. ייתכן ואף סביר להניח שתצטרכו להוסיף מתודות פרטיות (ורק פרטיות) **<u>נוספות</u>** עבור המחלקות המתוארות. חישבו היטב מה באמת צריך בכל מחלקה.

<u>כמובן, שגם אם לא נאמר, לכל המחלקות חובה להוסיף Constructor ו- Destructor דיפולטיבים (אפילו אם הוא ריק). כמו כן עליכם להוסיף מתודות set ו- get **לפי הצורך.**</u>

1. <u>מחלקת מספרים מרוכבים – Complex:</u>

מחלקה זו תייצג מספר מרוכב ותדע להתמודד עם פעולות אריתמטיות הנוגעות למספרים מרוכבים.

- <u>שדות:</u>
  - float – רכיב ממשי.
  - float – רכיב מדומה.
- <u>מתודות:</u>
  - getRad - מתודה שמחזירה את הרדיוס(float).
  - getPhase - מתודה שמחזירה את הזווית במעלות – degrees $\phi \in [0,360)$ (float).
    *עבור 0 הפאזה תוגדר להיות 0.

  <u>בנאים:</u>
  - אם אין קלט, יאותחל המספר כ – 0 (רכיב ממשי ורכיב מדומה שווים ל 0).
  - אם הקלט מכיל מספר אחד אזי המספר יאותחל כמספר ממשי (רכיב מדומה שווה ל 0).
  - אם הקלט מכיל שני מספרים אזי המספר יאותחל כמספר מרוכב.
- <u>העמסת אופרטורים</u>

  יש להעמיס את האופרטורים הבאים על מנת לבצע פעולות אריתמטיות בין מספרים שונים:
  - + - חיבור.
  - – - חיסור.
  - * - כפל.
  - / - חילוק.

  יש להעמיס בנוסף את האופרטורים הבאים
  - == - שוויון.
  - >> - אופרטור הדפסה.
  - ~ - אופרטור הצמוד (אופרטור אונארי).

2. <u>מחלקת – ComplexVec:</u>

מחלקה זו מכילה שדה מידע מסוג וקטור שמאחסן מספרים מרוכבים.

- <u>שדות:</u>
  - Vector – וקטור של אובייקט מסוג Complex (שהוגדר למעלה).

  וקטור הוא מערך דינאמי במובן שגודלו יכול להשתנות לאורך התוכנית, אין חובה להקצות אותו בצורה דינאמית כמובן. לקריאה נוספת על וקטורים:
  https://www.geeksforgeeks.org/vector-in-cpp-stl
  http://www.cplusplus.com/reference/vector/vector
- <u>מתודות:</u>
  - printElements - מתודה שמדפיסה את האיברים לפי הסדר שלהם ב vector.
  - insert - מתודה שמקבלת אובייקט מסוג complex ומכניסה אותו לווקטור.
- <u>העמסת אופרטורים</u>
  - [ ] - index operator – על מנת לפנות ל index ספציפי בשדה ה Vector של האובייקט.

3. מחלקת תפריט – Menu

מחלקה זו תנהל את התפריט. על מחלקה זו לבצע פעולות קלט/פלט מהמשתמש למעט שיטות שהוגדרו לכך במפורש במחלקות אחרות .יש להציג את התפריט בלולאה עד שהמשתמש יבחר לסיים אותה.

שדות:

ComplexVec – אובייקט אשר יכיל את המספרים המרוכבים המוזנים ע"י המשתמש.

שיטות:

mainMenu-שיטה המציגה את התפריט הבא:

```
========================================
<1> add a new complex number
<2> print all numbers
<3> +
<4> -
<5> /
<6> *
<7> ~
<8> polar coordinate
<9> exit
========================================
```

1. **הוספת מספר מרוכב חדש:** המשתמש יתבקש להזין רכיב ממשי ורכיב מדומה של המספר המרוכב. בעקבות הזנת הנתונים יוכנס לווקטור מספר מרוכב חדש. לדוגמא:

```
please insert complex number:
real: 2
imaginary: 2
```

2. **הדפסת כל המספרים המרוכבים:** פעולה זו תדפיס את כל המספרים המרוכבים שנמצאים בווקטור לפי סדר הכנסתם. לדוגמא עבור וקטור המכיל 2 מספרים מרוכבים:

```
1: 2 + 2i
2: 4 + 5.5i
```

3-6. **פעולות אריתמטיות:** בחירה זו תבקש מהמשתמש לבחור את המספרים המרוכבים שעליהם הוא רוצה לבצע את הפעולה ע"פ האינדקס שלהם בווקטור. במידה והמשתמש הזין אינדקסים שאינם בטווח המתאים, תודפס הודעה מתאימה. לדוגמא עבור פעולת חיבור:

```
index range start from 1; i.e., 1,2,3....
insert index of first number: -2
not in range, try again
2
insert index of second number: 1
(4 + 5.5i) + (2 + 2i) = 6 + 7.5i
```

7. **צמוד:** בחירה זו תבקש מהמשתמש לבחור את המספר המרוכב שעליו הוא רוצה לבצע את פעולת הצמוד. במידה והמשתמש הזין אינדקסים שאינם בטווח המתאים, תודפס הודעה מתאימה. לדוגמא:

```
index range start from 1; i.e., 1,2,3....
insert index of complex number: 2
The conjugate of 4 + 5.5i is 4 - 5.5i
```

8. **הצגה פולרית:** בחירה זו תאפשר למשתמש לפנות למספר מרוכב ע"י אינדקס מתאים על מנת להציגו בצורתו הפולרית. לדוגמא:

```
index range start from 1; i.e., 1,2,3....
insert index for printing its parameters: 1
```

phase(degree): 45
radius: 2.82843

9. **יציאה:** פעולה זו תסגור את התוכנית ותדפיס הודעה מתאימה.

** שימו לב – אתם נדרשים לחשוב על מקרי קצה ולטפל בהם. למשל במצב בו המשתמש ינסה להפעיל פעולה אריתמטית למרות שהווקטור ריק. במקרה כזה ניתן להדפיס הודעה מתאימה כמו למשל:

Complex Vector is empty

** במקרה בו המספר המרוכב מכיל חלק מדומה/חלק ממשי/גם וגם אשר שווים לאפס עליכם להציג בתצורה מלאה כולל אפסים. למשל:

1+0i
0+0i

ה main-

מייצרת אובייקט מסוג תפריט ומריצה את המתודה "תפריט ראשי" וזהו.