

תכנות מונחה עצמים**תרגיל 2**

מועד הגשה 11.8.2022 בשעה 23:50

הוראות הגשה:

1. הגשה באופן עצמאי בלבד. הגשה בקבוצות תוביל לציון 0 בעבודה.
2. אין לשתף או להעתיק את העבודה או חלקים ממנה. עבירה על הוראה זו תוביל לציון 0 בעבודה.
3. הגשה דרך מערכת מודול בלבד. שום עבודה לא מתקבלת במייל!
4. בעבודה זו יש לכתוב תכנית אחת עם 2 מחלקות. יש לכתוב עבור המחלקה קובץ cpp וקובץ h. בנוסף, יש לכתוב קובץ cpp עבור הפונקציה הראשית. סה"כ התכנית צריכה לכלול 3 קבצים. בנוסף, יש לענות בקובץ WORD או PDF על החלק התיאורטי. להגיש קובץ אחד בפורמט RAR או ZIP המכיל את כל הקבצים.
5. על משתני המחלקות להיות פרטיים.
6. להוסיף מילה const בכל מקום בקוד שהדבר אפשרי.
7. אין להשתמש בתרגיל זה בחומר שטרם נלמד בהרצאות. **אין להשתמש בהורשה.**

חלק א' – תיאורטי (16 נקודות)

1. (3 נקודות) האם ניתן לקרוא מתוך פונקציה סטטית של מחלקה לפונקציה של המחלקה שאינה סטטית? נמק!
2. (3 נקודות) האם ניתן להגדיר בתוך מחלקה שתי פונקציות בעלות אותו שם ואותם פרמטרים, שאחת מהן היא const והשניה לא? למשל, MyClass כוללת את שתי הפונקציות הבאות:

```
int MyClass::f(int a, int b) const;
int MyClass::f(int a, int b);
```
3. (4 נקודות) הפונקציה הבאה יוצרת אובייקט ממחלקה ClassA, ומחזירה מצביע אליו.

```
ClassA *func(int var1, int var2)
{
    ClassA obj(var1,var2);
    return &obj;
}
```

 - a. מה לא בסדר בפונקציה זו?
 - b. כתוב קוד תקין של פונקציה היוצרת אובייקט ממחלקה ClassA, ומחזירה מצביע אליו.
4. (6 נקודות) מחלקה MyClass מכילה את השדות הפרטיים הבאים:

```
class MyClass
{
private:
    int arr_size;
    static int count; //number of objects of this class
    float *arr;
public:
    ...
};
```

כתוב בנאי מעתיק ואופרטור = עבור MyClass.

חלק ב' – מעשי (84 נקודות)

הגדר מחלקה **Names** אשר תייצג קבוצה של שמות ללא חזרות. כל שם הוא מערך של לכל היותר 20 תווים ללא רווחים. יש להקצות לכל שם זיכרון דינמי באופן מדויק. המחלקה תכיל את המשתנים הבאים:

- **names** - מערך דו-מימדי של תווים שמוקצה באופן דינמי בגודל מדויק.
 - **size** – כמות השמות בקבוצה.
 - **total** – משתנה סטטי המכיל את סכום הגדלים של כל הקבוצות ($\text{size} = \text{סכום הגדלים של כל האובייקטים}$).
- בנוסף, יש להגדיר במחלקה את הפונקציות הבאות:
- **בנאי ללא פרמטרים** יוצר אובייקט שבו ערכו של **names** הוא **NULL**, ערכו של **size** הוא 0.
 - **בנאי מעתיק**.
 - **פונקציה הורסת**.
 - **אופרטור הדפסה** המדפיס את השמות של חברי הקבוצה.
 - **אופרטור +=** האופרטור מקבל כפרמטר שם. (מערך של תווים). אם הפרמטר לא נמצא בקבוצה, האופרטור מוסיף אותו לקבוצה. אחרת, האובייקט לא משתנה. על מנת להוסיף שם לקבוצה יש ליצור מערך גדול ב-1 מהמערך המקורי, להעתיק אליו את כל השמות שבמערך המקורי, וכן את הפרמטר. לבסוף יש לשחרר את המערך הישן. כמו כן יש להגדיל ב-1 את **size**.
 - **אופרטור -=** האופרטור מקבל כפרמטר שם. (מערך של תווים). אם הפרמטר נמצא בקבוצה, האופרטור משמיט אותו מהקבוצה. אחרת, האובייקט לא משתנה. על מנת להשמיט שם מהקבוצה יש ליצור מערך קטן ב-1 מהמערך המקורי, ולהעתיק אליו את כל השמות שבמערך המקורי חוץ מהפרמטר. לבסוף יש לשחרר את המערך הישן. כמו כן יש להקטין ב-1 את **size**. אם מוחקים את האיבר היחיד בקבוצה אז המצביע הופך ל-**NULL**.
 - **אופרטור >** מקבל כפרמטר אובייקט אחר מהמחלקה. מחזיר **true** אם הפרמטר מוכל באובייקט הנוכחי. (כל שם שבקבוצת הפרמטר נמצא גם באובייקט הנוכחי). אחרת, מחזיר **false**. האופרטור לא משנה את האובייקט.
 - **אופרטור <** מקבל כפרמטר אובייקט אחר מהמחלקה. מחזיר **true** אם הפרמטר מכיל את האובייקט הנוכחי. (כל שם שבאובייקט הנוכחי נמצא גם בפרמטר). אחרת, מחזיר **false**. האופרטור לא משנה את האובייקט.
 - **אופרטור ==** מקבל כפרמטר אובייקט אחר מהמחלקה. מחזיר **true** אם שני האובייקטים מכילים את אותם השמות. (לא בהכרח באותו הסדר). אחרת, מחזיר **false**. האופרטור לא משנה את האובייקט.
 - **אופרטור =** מקבל כפרמטר אובייקט אחר מהמחלקה. משנה את ערכו של האובייקט הנוכחי כך שיהיה שווה לפרמטר.
 - **אופרטור *** מקבל כפרמטר אובייקט אחר מהמחלקה. יוצר ומחזיר אובייקט המכיל את השמות המשותפים לשני האובייקטים. האופרטור לא משנה את האובייקט הנוכחי.
 - **אופרטור &&** מקבל כפרמטר שם. (מערך של תווים). מחזיר **true** אם הפרמטר נמצא בקבוצה. אחרת, מחזיר **false**. האופרטור לא משנה את האובייקט.
 - **פונקציה המחזירה** את סכומי הגדלים של כל הקבוצות. הפונקציה לא מדפיסה.

כתוב את הפונקציה הראשית בתכנית. הפונקציה הראשית מגדירה מערך דינמי של קבוצות בגודל שהמשתמש קובע. בתחילת התכנית כל הקבוצות ריקות. לאחר מכן, הפונקציה הראשית מציגה למשתמש את התפריט הבא עד שהוא בוחר לסיים את התכנית.

1 Print all 2 Add 3 Remove 4 Assign 5 Is equal 6 Is contain 7 Is contained 8 Intersection 9 Belong 10 Total size 11 Quit

אם המשתמש בוחר באפשרות 1 מודפסים כל האובייקטים שבמערך.

אם המשתמש בוחר באפשרות 2 הוא מתבקש להקליד מספר קבוצה ושם. אם מספר הקבוצה חוקי, מוסיפים את השם המוקלד לקבוצה. אחרת, מודפסת הודעת שגיאה.

אם המשתמש בוחר באפשרות 3 הוא מתבקש להקליד מספר קבוצה ושם. אם מספר הקבוצה חוקי, גורעים את השם המוקלד מהקבוצה. אחרת, מודפסת הודעת שגיאה.

אם המשתמש בוחר באפשרות 4 הוא מתבקש להקליד שני מספרי קבוצה. אם שני המספרים חוקיים, מבצעים השמה של הקבוצה הראשונה לשניה. אחרת, מודפסת הודעת שגיאה.

אם המשתמש בוחר באפשרות 5 הוא מתבקש להקליד שני מספרי קבוצה. אם שני המספרים חוקיים, התכנית בודקת אם הקבוצות שוות, ומדפיסה הודעה מתאימה. אחרת, מודפסת הודעת שגיאה.

אם המשתמש בוחר באפשרות 6 הוא מתבקש להקליד שני מספרי קבוצה. אם שני המספרים חוקיים, התכנית בודקת אם הקבוצה הראשונה מכילה את השניה, ומדפיסה הודעה מתאימה. אחרת, מודפסת הודעת שגיאה.

אם המשתמש בוחר באפשרות 7 הוא מתבקש להקליד שני מספרי קבוצה. אם שני המספרים חוקיים, התכנית בודקת אם הקבוצה הראשונה מוכלת בשניה, ומדפיסה הודעה מתאימה. אחרת, מודפסת הודעת שגיאה.

אם המשתמש בוחר באפשרות 8 הוא מתבקש להקליד 3 מספרי קבוצה. אם שלושת המספרים חוקיים, התכנית שמה בקבוצה הראשונה את החיתוך של הקבוצה השניה והשלישית. (הערך הישן של הקבוצה השלישית נמחק). אחרת, מודפסת הודעת שגיאה.

אם המשתמש בוחר באפשרות 9 הוא מתבקש להקליד שם. התכנית מדפיסה את המספרים של כל הקבוצות שמכילות שם זה.

אם המשתמש בוחר באפשרות 10 מודפס סכום הגדלים של כל הקבוצות.

אם המשתמש בוחר באפשרות 11 כל הזיכרון הדינמי משוחרר בצורה מסודרת, והתכנית מסתיימת.

דוגמא לריצת התכנית:

Enter number of groups: 5

1 Print all 2 Add 3 Remove 4 Assign 5 Is equal 6 Is contain 7 Is contained 8 Intersection 9 Belong 10 Total size 11 Quit

10

0

1 Print all 2 Add 3 Remove 4 Assign 5 Is equal 6 Is contain 7 Is contained 8 Intersection 9 Belong 10 Total size 11 Quit

2

Enter group num: 1

Enter name: Avi

1 Print all 2 Add 3 Remove 4 Assign 5 Is equal 6 Is contain 7 Is contained 8 Intersection 9 Belong 10 Total size 11 Quit

2

Enter group num: 1

Enter name: Beni

1 Print all 2 Add 3 Remove 4 Assign 5 Is equal 6 Is contain 7 Is contained 8 Intersection 9 Belong 10 Total size 11 Quit

2

Enter group num: 1

Enter name: Dorit

1 Print all 2 Add 3 Remove 4 Assign 5 Is equal 6 Is contain 7 Is contained 8 Intersection 9 Belong 10 Total size 11 Quit

2

Enter group num: 2

Enter name: Shira

1 Print all 2 Add 3 Remove 4 Assign 5 Is equal 6 Is contain 7 Is contained 8 Intersection 9 Belong 10 Total size 11 Quit

2

Enter group num: 2

Enter name: Avi

1 Print all 2 Add 3 Remove 4 Assign 5 Is equal 6 Is contain 7 Is contained 8 Intersection 9 Belong 10 Total size 11 Quit

1

Group 0:

Group 1: Avi Beni Dorit

Group 2: Shira Avi

Group 3:

Group 4:

1 Print all 2 Add 3 Remove 4 Assign 5 Is equal 6 Is contain 7 Is contained 8 Intersection 9 Belong 10 Total size 11 Quit

4

Enter source: 1

Enter destination: 4

1 Print all 2 Add 3 Remove 4 Assign 5 Is equal 6 Is contain 7 Is contained 8 Intersection 9 Belong 10 Total size 11 Quit

1

Group 0:

Group 1: Avi Beni Dorit

Group 2: Shira Avi

Group 3:

Group 4: Avi Beni Dorit

1 Print all 2 Add 3 Remove 4 Assign 5 Is equal 6 Is contain 7 Is contained 8 Intersection 9 Belong 10 Total size 11 Quit

9

Enter name: Beni

1 4

1 Print all 2 Add 3 Remove 4 Assign 5 Is equal 6 Is contain 7 Is contained 8 Intersection 9 Belong 10 Total size 11 Quit

8

Enter two source groups: 1 2

Enter destination group: 0

1 Print all 2 Add 3 Remove 4 Assign 5 Is equal 6 Is contain 7 Is contained 8 Intersection 9 Belong 10 Total size 11 Quit

1

Group 0: Avi

Group 1: Avi Beni Dorit

Group 2: Shira Avi

Group 3:

Group 4: Avi Beni Dorit

1 Print all 2 Add 3 Remove 4 Assign 5 Is equal 6 Is contain 7 Is contained 8 Intersection 9 Belong 10 Total size 11 Quit

10

9

1 Print all 2 Add 3 Remove 4 Assign 5 Is equal 6 Is contain 7 Is contained 8 Intersection 9 Belong 10 Total size 11 Quit

3

Enter group: 4

Enter name: Beni

1 Print all 2 Add 3 Remove 4 Assign 5 Is equal 6 Is contain 7 Is contained 8 Intersection 9 Belong 10 Total size 11 Quit

1

Group 0: Avi

Group 1: Avi Beni Dorit

Group 2: Shira Avi

Group 3:

Group 4: Avi Dorit

1 Print all 2 Add 3 Remove 4 Assign 5 Is equal 6 Is contain 7 Is contained 8 Intersection 9 Belong 10 Total size 11 Quit

7

Enter left group: 3

Enter right group: 0

Group 3 is contained in group 0.

1 Print all 2 Add 3 Remove 4 Assign 5 Is equal 6 Is contain 7 Is contained 8 Intersection 9 Belong 10 Total size 11 Quit

11

Goodbye

עבודה פוריה !!!