תרגילים 7

ממשו את הפונקציות הבאות: (תרגישו בנח לדלג בין תרגילים)

First run:
git **clone** https://github.com/jonmuehlst/workspace_a.git

Otherwise:
git **pull** https://github.com/jonmuehlst/workspace_a.git

1. אתחול דינמי של מערך דו מימדי ע"י new
    1.1.1. קלט: מספר השורות , מספר העמודות
    1.1.2. פלט: מצביע למצביע (מערך דו מימדי דינמי)

```
int ** initMatrix(size_t,size_t);
```

2. שחרור זיכרון (שהוקצה דינמית) עבור מערך דו מימדי ע"י delete
    2.1. מערך דו מימדי (ע"י מצביע למצביע), מספר השורות, מספר העמודות
    2.2. רמז: שחרור זיכרון למערך מבצעים ע"י delete עם סוגריים מרובעים []

```
void deleteMatrix(int**,size_t,size_t);
```

3. פונקציה שסופרת "פרחים" במטריצה .
    3.1. נגדיר פרח ע"י המטריצה (לדוגמה בלבד) הבאה:

11	מספר כלשהו	1
מספר כלשהו	20	מספר כלשהו
5	מספר כלשהו	3

כאשר הערך באמצע (20) שווה לסכום ארבעת ה-"עלים" שלו. מצאנו פרח

    3.2. קלט: מערך דו מימדי (ע"י מצביע למצביע), מספר השורות , מספר העמודות
    3.3. פלט: מספר הפרחים במטריצה

```
int sumFlowers(int ** a,size_t,size_t);
```

    3.4. כתבו פונקציית עזר בוליאנית שמבודקת אם אינדקס מסוים מקיים את ההגדרה
        3.4.1. קלט: מערך דו מימדי (ע"י מצביע למצביע), אינדקס שורה , אינדקס עמודה
        3.4.2. פלט: 1 אם ההגדרה מתקיימת \ 0 במקרה שלא

```
int isFlower( int ** a,size_t,size_t);
```

4. כפל מטריצות
    4.1. קלט: שלוש מטריצות ריבועיות A,B,C, מימד (מספר השורות \ עמודות)

4.2. פלט: ההכפלה של המטריצות A ו-B תישמר במטריצה C

```
void multiplyMatrix(int ** ,int ** ,int **,size_t);
```