דוח מעבדה 1

בקורס מעבדה לבינה מלאכותית

הגשת:

מוחמד סח – 209332212

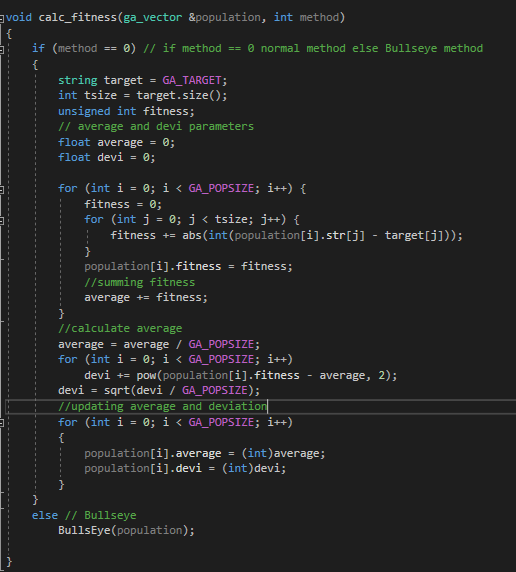
יוסף עאסלה - 207344573

חלק א':

סעיף 1:

מוצע הFITNESS – של האוכלוסיה ושל סטיית התקן מהממוצע:

הוספנו קוד לפונקצית calc\_fitnessהנתונה :



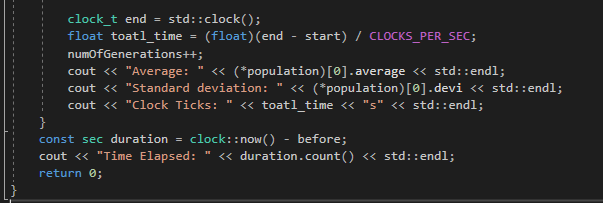
והוספנו גם average ,devi(deviation) ל ga\_struct.

סעיף 2:

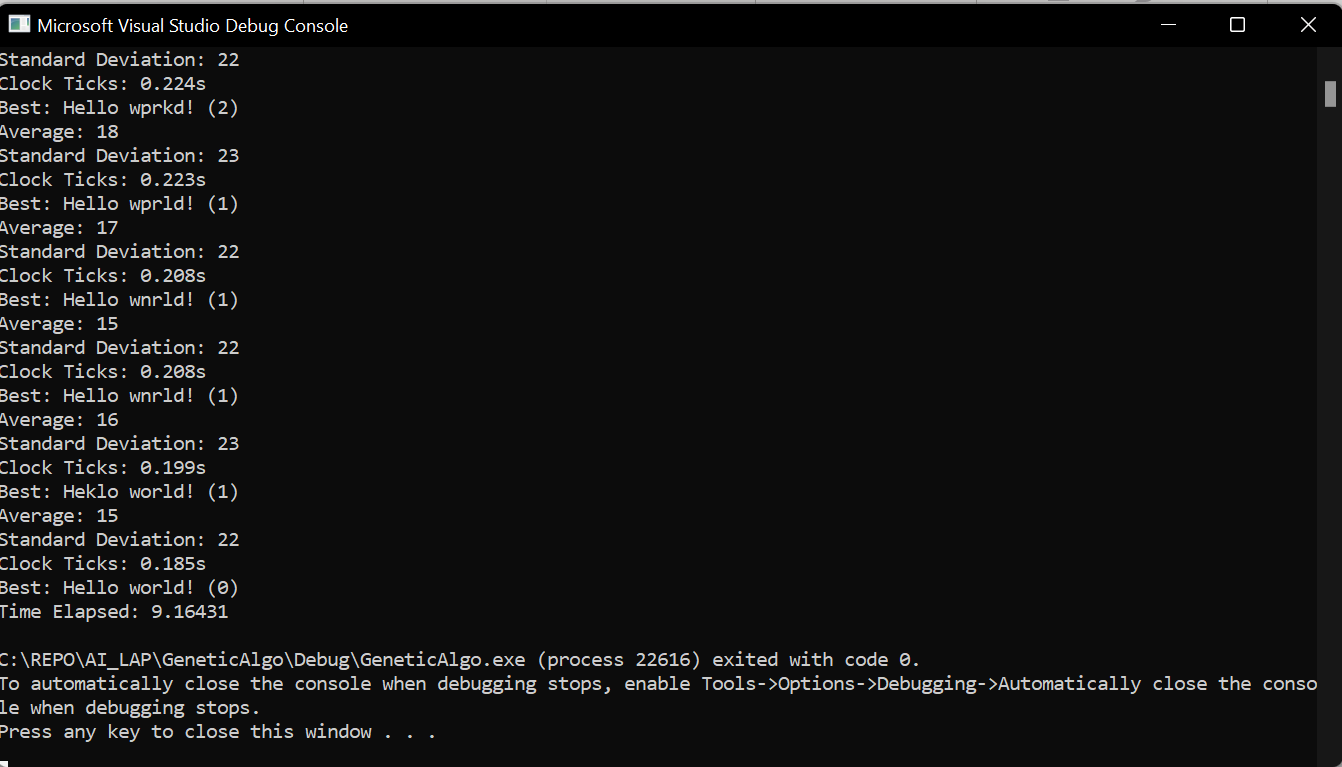
זמו ריצה , clock ticks ו זמו ריצה אבסולוטי:

השתשנו בספריות chrono ו ctime , הוספנו ל main function :





דוגמא להרצת קוד:

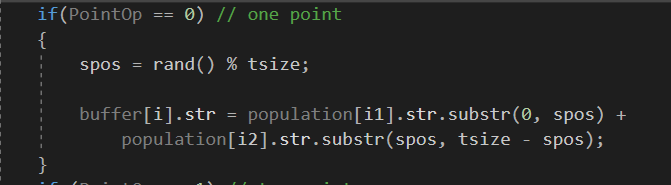


סעיף 4:

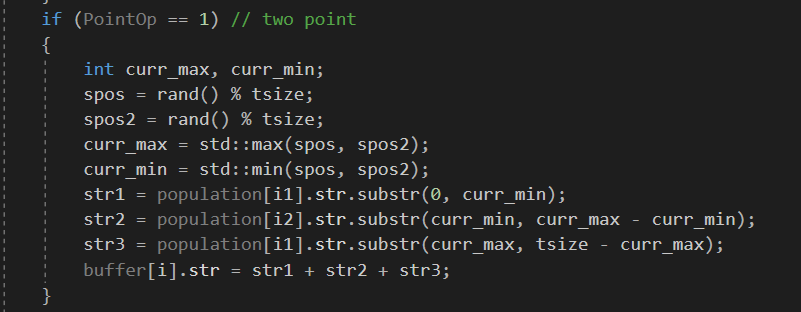
אופרטורים לשיחלוף:

הוספנו קטע קוד ל mate function שקיבלנו:

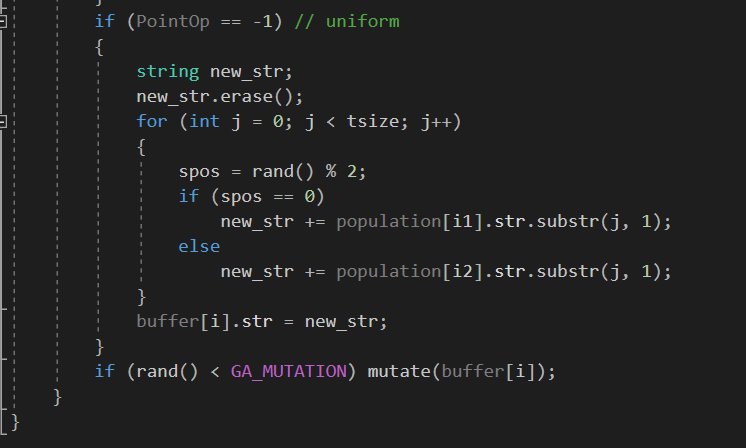
מה שקבלנו הוא ה one point :



הוספנו את ה two point :



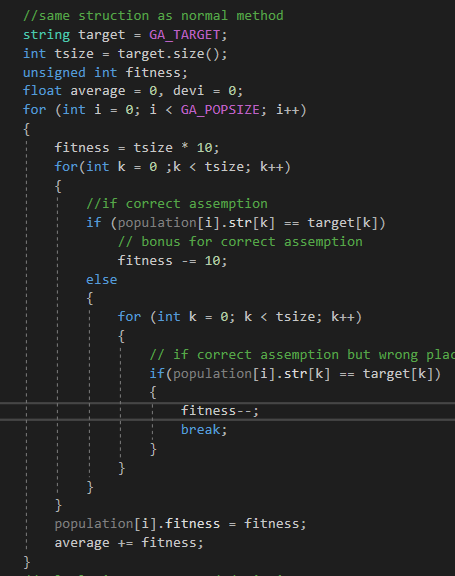
וגם הuniform :



סעיף 5:

היוריסטיקה נוספת "בול פגיעה":

הוספנו היוריסטיקה חדשה "בול פגיעה" שהיא נותנת משקל נוסף על ניחוש נכון:



סעיף 6:

השוואה בין ההיוריסטיקות:

הרצנו את הקוד מספר פעמים ,כמו שרואים בגרף היוריסטיקת "בול פגיעה יותר מהירה וזה בגלל שהיא נותנת משקל נוסף לחיזוי נכון.

עבור uniform point operation :

עבור point operation1 :

עבור 2 point operation:

סעיף 7:

point operations ו mate function הם החלק האחראי על ה EXPLORATION ,מפנה שבחלק זה אנחנו "מערבבים" גינים ו מקבלים גינים חדשים.

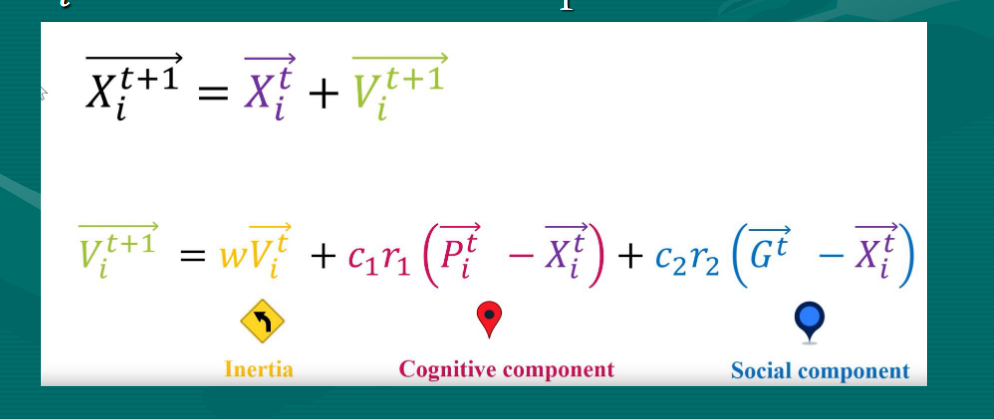
פונקצית elitisim היא החלק האחראי על ה EXPLOITATION מפנה שבחלק זה אנחנו מעבירים אחוז מסויים מהגינים הכי טובים לדור הבא.

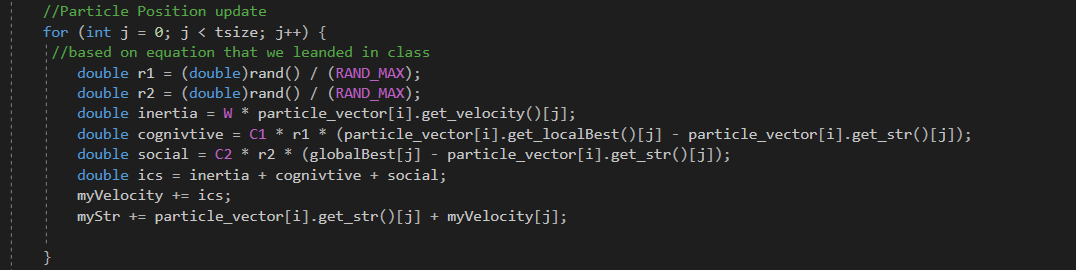
סעיף 8:

PSO ALGORITHIM

הוספנו מחלקה חדשה PSO שתומכת ב particles

השתמשנו בנוסחה שלמדנו בקורס המבוא :





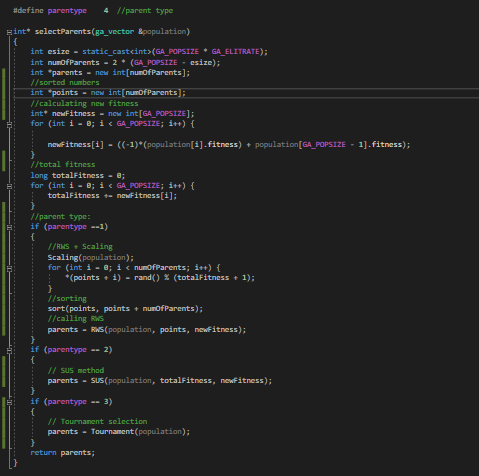
סעיף 9:

נשתמש בהיוריסטיקה " פול פגיעה" ו ב 1 point operation בהשוואה זו:

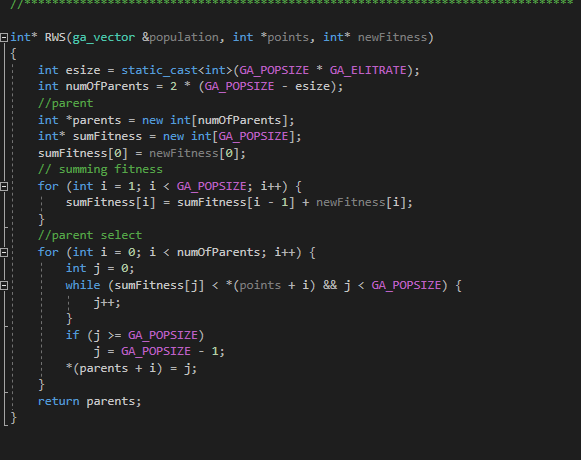
כמו שאנחנו רואים בגרף יש הבדל גדול בין זמן הריצה ו מספר ה איטרציות בין שני האלגוריתמים. האלגוריתם הגנטי יותר מהיר ו יעיל לכן הוא מועדף יותר לשימיש.

חלק ב:

סעיף 1: הוספנו selectparent שהיא אחראית על בחירת התמיכה בשיטות הבחירה



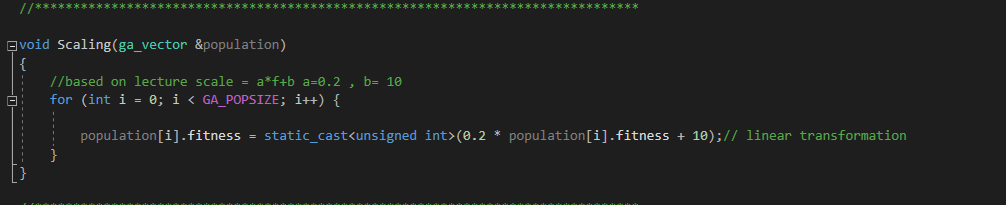
RWS:



SCALING:

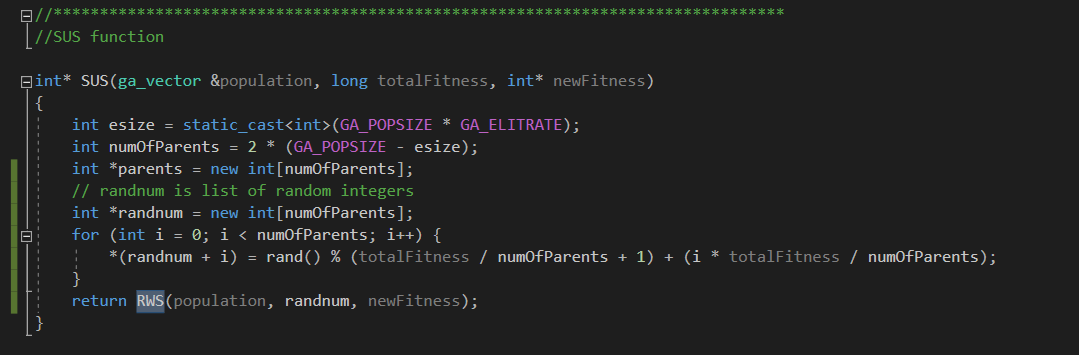
השתמשנו בנוסחה שלמדנו בהרצאה a\*f+b , קבענו:

a=0.2 , b= 10



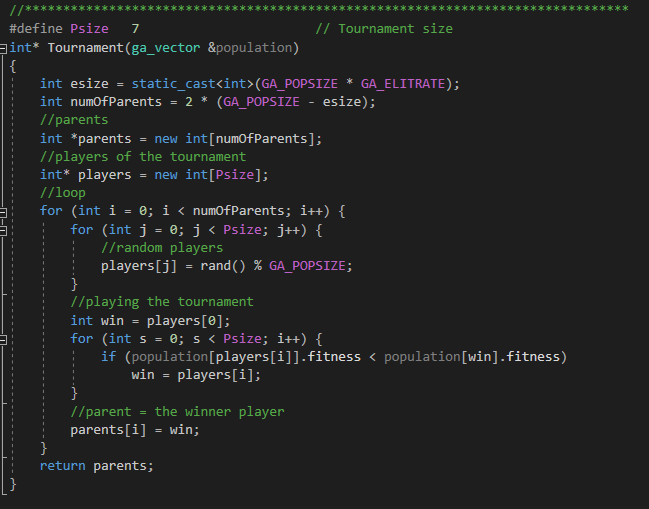
SUS:

השתמשנו באלגוריתם שפותר בעזרת RWS



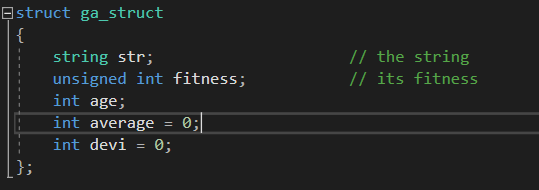
TOURNAMENT:

השתמשנו באלגוריתם שלמדנו בהרצאה: נבחר מספר מועמדים לטורניר psize נתחיל עם המועמד הראשון ו אם נמצא מועמד יותר טוב בתהליך נמשיך אותו את הטורניר.

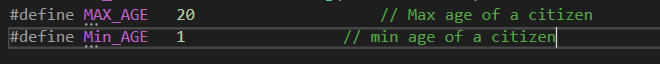


סעיף 2:

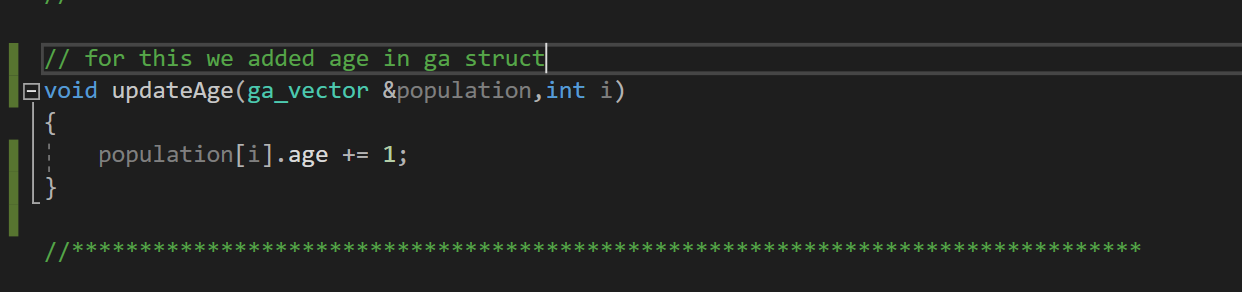
הוספנו ל GA\_struct את ה age של כל citizen



כמו כן הוספנו ב define את הגיל המקסימלי ו המינמאלי

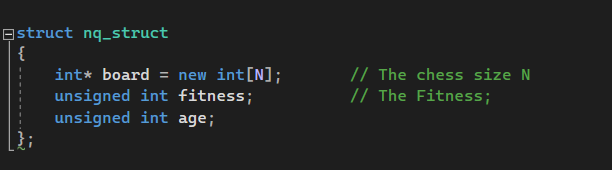


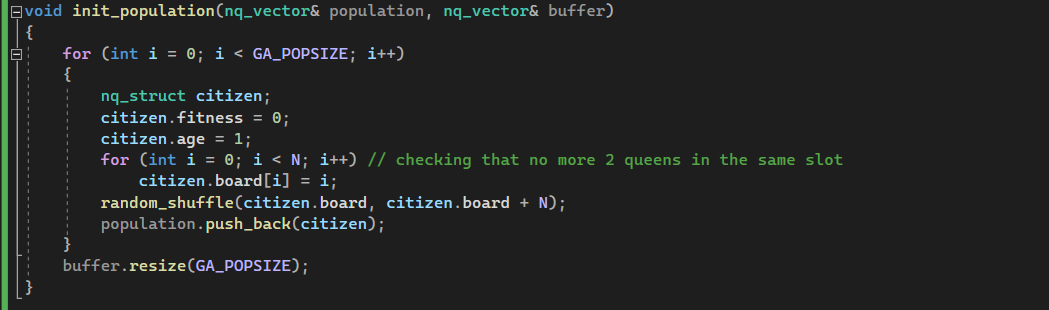
הגדרנו פינקציה שמעדכנת את הגיל



סעיף 3:

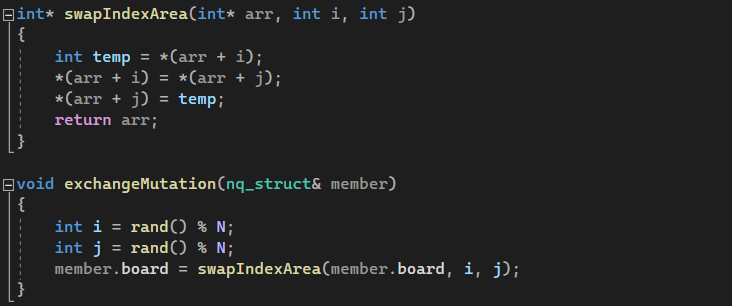
עשינו שינוי על ה nq\_struct כך שיהיה לנו מערך באורך מספר המלכות שמהוות פרמוטציה כלשהו, כלומר לא יהיו שתי מלכות באותה עמודה או שורה.

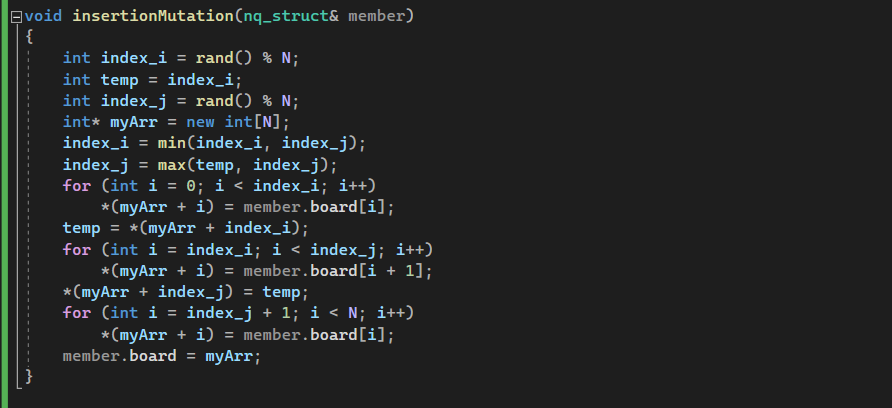




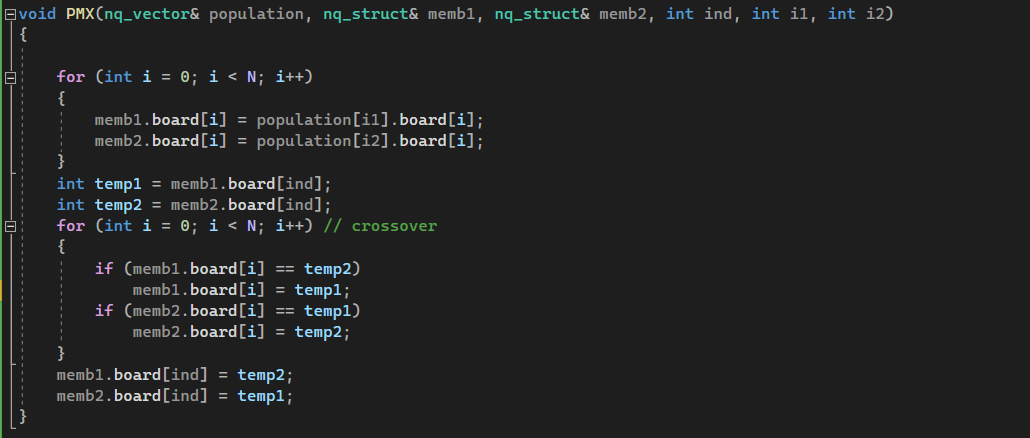
סעיף 4:

אופרטורי מוטציה:



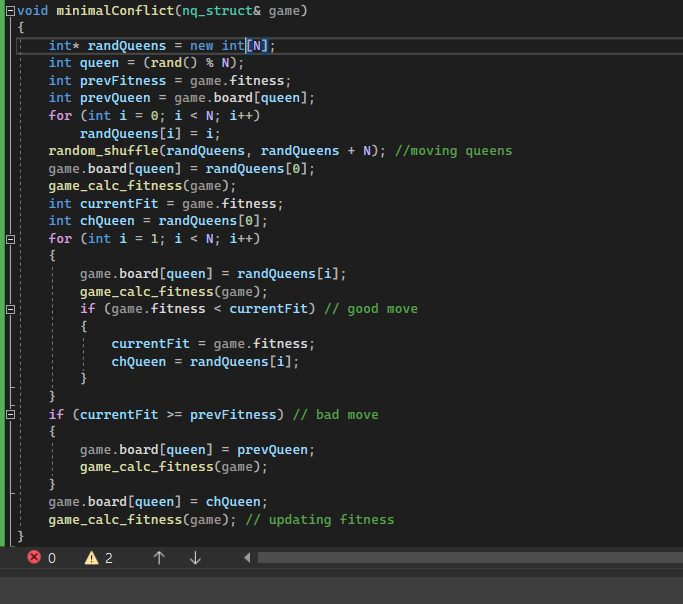


אופרטורי שחלוף:



סעיף 6:

Minimal Conflict Algorithm:



אנחנו נשתמש בפונקציה הזו כדי לתקן את הלוח על ידי הזזת המלכה למקום שבו פחות איומים.

אחרי שהרצנו את הקוד כמה פעמיים, מצאנו שאלגוריתמים ה N Queen עבור genetic algorithm יעיל ומהיר יותר, מכך שאלגוריתם Minimal Conflict עצר ברוב האיטרציות אחרי כניסה ל local opt. לכן עדיף להשתמש ב genetic algorithm.