

תכנות מתקדם – 150024**תרגיל בית מספר 9****STL****שים/י לב:**

- א. הקפד/י על קריאות התכנית ועל עימוד (Indentation).
- ב. הקפד/י לבצע בדיוק את הנדרש בכל שאלה.
- ג. בכל אחת מהשאלות יש להגדיר פונקציות במידת הצורך עבור קריאות התכנית.
- ד. יש להגיש את התרגיל על פי ההנחיות להגשת תרגילים (המופיע באתר הקורס) וביניהם:
השתמש/י בשמות משמעותיים עבור המשתנים.
יש לתעד את התכנית גם עבור פונקציות אותם הנך מגדיר/ה וכן על תנאים ולולאות וקטעי קוד מורכבים, ובנוסף, **דוגמת הרצה לכל תכנית בסוף הקובץ!**
הגשה בזוגות.

הערה חשובה: לכל תרגיל בית מוגדר שבוע אחד בלבד להגשה, אלא אם כן קיבלת הוראה אחרת מהמרצה שלך. תיבות ההגשה הפתוחות לא מהוות היתר להגשה באיחור.

שאלה מס' 1:

- בתרגיל זה עוסקים במערך הצל"שים לאחר מבצע שומר החומות.
- במבצע שומר החומות השתתפו טירונים, מפקדים וקצינים.
- נקבעו קריטריונים לקבלת צל"ש עבור כל תפקיד:
- טירון: השתתף בלפחות 10 מבצעים צבאיים וממוצע הערכות של השתתפותו בכל מבצע גבוה מ-95
 - מפקד: השתתף בלפחות 7 מבצעים צבאיים וממוצע הערכות של השתתפותו בכל מבצע גבוה מ-90, ובנוסף הוא קרבי
 - קצין: השתתף ביותר מ-2 מבצעים צבאיים וקיבל ציון סוציומטרי מהחיילים של לפחות 92

על מנת לייצג תפקידים אלו, עליך להגדיר היררכיה של מחלקות באופן הבא:

- א. מחלקה אבסטרקטית בסיסית בשם Soldier לייצוג חייל.
- תכונות המחלקה: מספר זהות, שם פרטי, שם משפחה, מס' המבצעים בהם השתתף החייל.
- הוסף/י למחלקה את המתודות הבאות:
- constructor לאתחול התכונות
 - virtual destructor
 - medal() מתודה בוליאנית הבודקת האם החייל זכאי לצל"ש
 - print() עבור הדפסת נתוני החייל. את נתוני החייל יש להתאים בהתאם לתכונותיו.
- כאשר יש להדפיס תחילה את שם התכונה ולאחר מכן את ערכי התכונה, כל תכונה בשורה נפרדת. (שים/י לב: כשמדובר על התכונה של ציונים למבצעים, במידה שאין מבצעים אין להדפיס את המילה grades).

- string soldierType() מחרוזת המציינת מהו הטיפוס של החייל. (המתודה תחזיר "private", "commander", "officer" בהתאמה לטיפוסי החיילים השונים)

- מחלקה PrivateSoldier היורשת מ-Soldier עבור ייצוג טירון, המכילה, בנוסף לפרטים האישיים, גם מצביע למערך ציוני הערכות של המבצעים הצבאיים בהם השתתף (טווח הציונים 0-100). שים לב, מדובר במחלקה עם שטחים דינאמיים המחייבת יישום של העתקה עמוקה במתודות הרלוונטיות.
 - מחלקה Commander היורשת מ-PrivateSoldier עבור ייצוג מפקד, המכילה, בנוסף לפרטים הקיימים, תכונה בוליאנית המציינת האם המפקד קרבי או לא
 - מחלקה Officer היורשת מ-Soldier עבור ייצוג קצין, המכילה, בנוסף לפרטים האישיים, תכונה עבור ציון הסוציומטרי שלו.
- שים לב, עליך להגדיר את המחלקות בצורה נכונה, הגדרת בנאים נכונים (כולל בנאי העתקה ובנאי הזזה), הגדרת פונקציות החזרה (get), (אין צורך להגדיר פונקציות הצבה (set)), הגדרת מתודות כוירטואליות, כוירטואליות טהורות וכדו' אופן הדפסת הנתונים עבור כל חייל מתואר באופן הבא:

PrivateSoldier ID ת.ר. שם פרטי: first name שם משפחה: last name מס' מבצעים: num operations ובמידה שמספר מבצעים גדול מ-0 הדפסת מערך ציוני הערכות עם: grades רווח בין ציון לציון	Commander ID ת.ר. שם פרטי: first name שם משפחה: last name מס' מבצעים: num operations ובמידה שמספר מבצעים גדול מ-0 הדפסת מערך ציוני הערכות עם: grades רווח בין ציון לציון combat: yes/no	Officer ID ת.ר. שם פרטי: first name שם משפחה: last name מס' מבצעים: num operations ציון סוציומטרי: sociometric score
---	--	--

- בתכנית הראשית, נגדיר ווקטור vector או רשימה list (לבחירתך) בעזרת STL container אשר יכול להכיל חיילים **מכל** הסוגים. בנוסף, עליך להגדיר מספר פונקציות גלובאליות הפועלות על ווקטור או רשימה של חיילים (מכל הסוגים) באופן הבא:
 - פונקציה add המקבלת ווקטור או רשימה (של STL) של חיילים ומוסיפה לתוכנו חייל חדש.

לשם מימוש הפונקציה יש להשתמש בפלטים הבאים בלבד:

```
cout << "choose a soldier\n";
cout << "enter 1 to add a private\n";
cout << "enter 2 to add a commander\n";
cout << "enter 3 to add an officer\n";

cout<<"enter id, first name, last name and number of operations\n";
cout<<"enter "<<numOfOperations<<" grades\n";
cout<<"enter 1 if the soldier is combat and 0 if not\n";
cout<<"enter the sociometric score\n";
```

- פונקציה medal המקבלת ווקטור או רשימה של חיילים ומדפיסה מתוכה רק את הפרטים של החיילים הזכאים לצל"ש.

פונקציה highestSociometricScore המקבלת ווקטור או רשימה של חיילים ומחזירה מצביע לחייל ברשימה אשר תפקידו קצין וציון הסוציומטרי שלו הגבוה ביותר. במידה ואין קצינים ברשימה הפונקציה תחזיר NULL.

הערה חשובה לצורך מימוש פונקציה יש לקרוא לפונקציה getSocioMetric עבור המחלקה Officer. כיוון שרשימת החיילים נמצאת במבנה המכיל אובייקטים מסוג Soldier תקבלו שגיאת קומפילציה (אין אפשרות לזמן מתודות שאינן מוגדרות במחלקה Soldier). כדי לעקוף את הבעיה הזו:

- לפני הזמנת המתודה של getSocioMetric יש לבדוק שהחייל הוא מסוג Officer
- עושים המרה לאובייקט כך שהקומפילר יזהה שניתן להפעיל על העצם מתודה השייכת למחלקה Officer לדוגמה:
עבור ההגדרה הבאה:

```
Soldier *soldier = new Officer();
```

כותבים:

```
if (soldier->soldierType()=="officer")  
    ((Officer *) soldier)->getSociometric();
```

כאמור, בתכנית הראשית יש להגדיר ווקטור vector או רשימה list (לבחירתך) בעזרת STL container אשר יכול להכיל חיילים **מכל** הסוגים (טירון, מפקד, קצין). לאחר מכן, יוצג למשתמש בחירת פעולה לביצוע (בלולאה) עד לבחירת STOP.

הפעולות האפשריות:

- 0 – סיום התוכנית
- 1 – הוספת חייל חדש
- 2 – הדפסת נתוני החיילים הזכאים לצל"ש
- 3 – הדפסת שם (משפחה ופרטי) של החייל בתפקיד קצין בעל ציון סוציומטרי גבוה ביותר
- 4 – הדפסת מספר החיילים הטייונים הזכאים לצל"ש
- 5 – הדפסת שמות (משפחה ופרטי) החיילים המפקדים שאינם בקרבי
- 6 – הדפסת הודעה האם קיים ברשימה חייל כלשהו שהשתתף ביותר מ-15 מבצעים
- 7 – מחיקה מהווקטור/מהרשימה חיילים קצינים שלא השתתפו כלל במבצעים

נתונה התכנית הראשית הבאה, עליך להשלים את שורות הקוד החסרות שים' לב, בכל מקום המסומן בקו תחתי יש להשלים שורה אחת בלבד (לא בהכרח לפי אורך הקו) יש להיעזר בביטויי lambda ולקרוא לפונקציות האלגוריתמיות המוגדרות בספריה algorithm של STL.

הערה: בשאלה זו ניתן להיעזר בפירוט על STL המופיע באתר הבא:

<https://www.cplusplus.com/reference/>

```
#include <iostream>
#include <list>
#include <vector>
#include <algorithm>
#include <string>
using namespace std;

enum option {
    EXIT, // סיום התוכנית
    ADD_NEW_SOLDIER, // הוספת חייל חדש
    DESERVES_MEDAL, // הדפסת פרטי כל החיילים הזכאים לצל"ש
    HIGHEST_SOCIOMETRIC, // הדפסת שם (משפחה ופרטי) של החייל בעל ציון סוציומטרי מקסימלי
    PRIVATE_MEDAL_COUNT, // הדפסת מספר החיילים הטירונים הזכאים לצל"ש
    NONCOMBAT_COMMANDER, // הדפסת שמות (משפחה ופרטי) של המפקדים שאינם בקרבי
    SUPER_SOLDIER, // הדפסת הודעה מתאימה, האם קיים חייל שהשתתף ביותר מ-15 מבצעים צבאיים
    REMOVE_OFFICER, // מחיקת כל החיילים הקצינים שלא השתתפו כלל במבצע צבאי
};

void add(______); // השלם: פרמטר- ווקטור או רשימה
void printMedalList(______); // השלם: פרמטר- ווקטור או רשימה
Soldier* highestSociometricScore (______); // השלם: פרמטר- ווקטור או רשימה

int main()
{
    Soldier* s;
    _____ // הצהרה על ווקטור או רשימה של חיילים

    int op;
    cout<<"enter 0-7\n";
    cin>>op;
    while(op!=EXIT)
    {
        switch (op)
        {
            case ADD_NEW_SOLDIER: add( _____ ); // הוספת חייל חדש

            break;

            case DESERVES_MEDAL: printMedalList( _____ ); // הדפסת פרטי הזכאים לצל"ש
            break;

            case HIGHEST_SOCIOMETRIC: // הדפסת שם הקצין בעל ציון סוציומטרי גבוה ביותר
                s=highestSociometricScore( _____ ); // השלם: פרמטר- ווקטור או רשימה
                cout<<"Officer with the highest sociometric score: ";
                cout <<s->getFirstName()<<' '<<s->getLastName()<<endl;

                break;
        }
    }
}
```

```

        case PRIVATE_MEDAL_COUNT: cout<<"number of privates that received
medals: ";
        _____ //הדפסת מספר הזכאים לצל"ש בטירונים
        cout<<endl;
        break;
        case NONCOMBAT_COMMANDER: cout << "list of noncombat commanders: ";
        //הדפסת רשימת(שם משפחה ופרטי) החיילים המפקדים שאינם בקרבי/
        _____
        cout<<endl;
        break;
        case SUPER_SOLDIER:
            if( _____ ) //קיים חייל שהשתתף יותר מ-15 מבצעים צבאיים

                cout << "there is at least one soldier that did more than 15 operations\n";
            else
                cout << "no soldier did more than 15 operations\n";
            break;
        case REMOVE_OFFICER://מחיקה מהווקטור/רשימה של כל החיילים קצינים שאינם השתתפו כלל
        במבצעים צבאיים
        _____
        _____ //הדפסת כל הרשימה לאחר מחיקת האיברים
        break;
    };
    cout<<"enter 0-7\n";
    cin>>op;
}
return 0;
}

```

דוגמה להרצה התכנית

```
enter 0-7
1
choose a soldier
enter 1 to add a private
enter 2 to add a commander
enter 3 to add an officer
1
enter id, first name, last name and number of operations
111 aaa aaa 3
enter 3 grades
100 95 98
enter 0-7
1
choose a soldier
enter 1 to add a private
enter 2 to add a commander
enter 3 to add an officer
2
enter id, first name, last name and number of operations
222 bbb bbb 0
enter 1 if the soldier is combat and 0 if not
1
enter 0-7
1
choose a soldier
enter 1 to add a private
enter 2 to add a commander
enter 3 to add an officer
3
enter id, first name, last name and number of operations
333 ccc ccc 0
Enter the sociometric score
100
enter 0-7
3
Officer with the highest sociometric score: ccc ccc
enter 0-7
4
number of privates that received medals: 0
enter 0-7
5
list of noncombat commanders :
```

```
enter 0-7
6
no soldier did more than 15 operations
enter 0-7
7
private
ID: 111
first name: aaa
last name: aaa
num operations: 3
grades: 100 95 98
commander
ID: 222
first name: bbb
last name: bbb
num operations: 0
combat: yes
enter 0-7
0
```

בהצלחה רבה!!