

סדנא ב- ++C – 150018

תרגיל בית מספר 10

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 יחידנית - אין להגיש 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.

- ב. מחלקה Private 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.
- ג. מחלקה Commander היורשת מ-Private עבור ייצוג מפקד, המכילה, בנוסף לפרטים הקיימים, תכונה בוליאנית המציינת האם המפקד קרבי או לא
- ד. 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 מתודות כוירטואליות, כוירטואליות טהורות וכדו'
אופן הדפסת הנתונים עבור כל חייל מתואר באופן הבא:

<u>Private</u>
ID ת.ז.
שם פרטי: first name
שם משפחה: last name
מס' מבצעים: num operations
הדפסת מערך ציוני הערכות עם: grades
רווח בין ציון לציון

<u>Commander</u>
ID ת.ז.
שם פרטי: first name
שם משפחה: last name
מס' מבצעים: num operations
הדפסת מערך ציוני הערכות עם: grades
רווח בין ציון לציון
combat: yes/no

<u>Officer</u>
ID ת.ז.
שם פרטי: first name
שם משפחה: last name
מס' מבצעים: num operations
ציון סוציומטרי: sociometric score

- ה. הוסף לכל אחת מהמחלקות הנ"ל מתודה בשם `string soldierType()` המחזירה מחרוזת המציינת מהו הטיפוס של החייל. (המתודה תחזיר "private", "commander", "officer" בהתאמה לטיפוסי החיילים השונים)
- ו. 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<<"enter 1 to add a private soldier\n";
cout<<"enter 2 to add a commander soldier\n";
cout<<"enter 3 to add a officer soldier\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 number of sociometric score\n";
```

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

- פונקציה `mostSociometricScore` 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 זו יש להגדיר מתודה המחזירה את הציון הסוציומטרי של הקצין והיא קיימת רק במחלקה `Officer`. כיוון שהחיילים נתונים במבנה המכיל אובייקטים מסוג `Soldier` אין אפשרות לזמן מתודות שאינן מוגדרות במחלקה `Soldier`. ולכן יש להוסיף מתודה וירטואלית כזו למחלקה `Soldier`. המתודה לא תוגדר כוירטואלית טהורה בכדי לא לחייב את כל הבנים לממש אותה. במידה והיא לא אמורה להיקרא היא תזרוק שגיאה:
"ERROR: this function is just for officer soldier"

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 הטירונים 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/י לב, בכל מקום המסומן בקו תחתי יש להשלים שורה אחת בלבד (לא בהכרח לפי אורך הקו) יש להיעזר בביטוי `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{
    stop,                // סיום התוכנית
    addNewSoldier,        // הוספת חייל חדש
    medalList,            // הדפסת פרטי כל החיילים הזכאים לצל"ש
    mostSociometric,      // הדפסת שם (משפחה ופרטי) של החייל בעל ציון סוציומטרי מקסימלי
    countMedalPrivate,    // הדפסת מספר החיילים הטירונים הזכאים לצל"ש
    noCombatCommander,   // הדפסת שמות (משפחה ופרטי) של המפקדים שאינם בקרבי
    overSoldier,          // הדפסת הודעה מתאימה, האם קיים חייל שהשתתף ביותר מ 15 - מבצעים צבאיים
    removeOfficer,        // מחיקת כל החיילים הקצינים שלא השתתפו כלל במבצע צבאי
};

void add(_____); // השלם! פרמטר- ווקטור או רשימה
void medal(_____); // השלם! פרמטר- ווקטור או רשימה
Soldier* mostSociometricScore (_____); // השלם! פרמטר- ווקטור או רשימה

int main()
{
    _____ // הצהרה על ווקטור או רשימה של חיילים

    int op;
    cout<<"enter 0-7\n";
    cin>>op;
    while(op!=stop)
    {
        switch (op)
        {
            case addNewSoldier:add( _____ ); // הוספת חייל חדש
                break;
            case medalList:medal( _____ ); // הדפסת פרטי הזכאים לצל"ש
                break;
        }
    }
}

```

```

case mostSociometric: // הדפסת שם הקצין בעל ציון סוציומטרי גבוה ביותר
    Soldier* s=mostSociometricScore( ____ ); //השלם פרמטר-וקטור או רשימה
    cout<<"Officer soldier with most sociometric score: ";
    cout <<s->getFname()<<' '<<s->getName()<<endl;

    break;

case countMedalPrivate: cout<<"# private soldier for medal: ";
    _____ //הדפסת מספר הזכאים לצל"ש בטירונים
    cout<<endl;
    break;

case noCombatCommander: cout << "list of no combat commander soldier :";
    _____
    cout<<endl;
    break;

case overSoldier:
    if( _____ ) // קיים חייל שהשתתף יותר מ-15 מבצעים צבאיים
        cout<<"there is a soldier that takes more than 15
operations\n";

    else cout<<"no soldier takes more than 15 operations\n";
    break;

case removeOfficer:
    _____
    _____ //הדפסת כל הרשימה לאחר מחיקת האיברים
    break;

};
cout<<"enter 0-7\n";
cin>>op;

}
return 0;

} }

```

דוגמה להרצה התכנית

enter 0-7

1

enter 1 to add a private soldier

enter 2 to add a commander soldier

enter 3 to add a officer soldier

1

enter id. first name. last name and number Of operations

enter 0-7

6

no soldier takes more than 15 operations

enter 0-7

7

private

ID: 111

first name: aaa

last name: aaa

num operations: 3

grades: 100 95 98

בהצלחה רבה!!