

סדנא ב- C++ – 150018**תרגיל בית מספר 8****תור ומחסנית****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.

שים/י לב, בתרגיל זה מומלץ להשתמש ב-template אך לא חובה.
בנוסף, אין להשתמש ב-STL!

שאלה מס' 1:

בשאלה זו נגדיר תכנית אשר תקלוט מהמשתמש ביטוי מתמטי (infix) **כמחרוזת**, והתכנית תחשב את ערך הביטוי תוך שמירה על סדר פעולות חשבון. נעשה זאת באמצעות שימוש במחסנית.

- א. הגדר/י פונקציה גלובאלית בשם infixToPostfix המקבלת כמחרוזת ביטוי בייצוג infix ומחזירה מחרוזת חדשה עם ביטוי זהה בייצוג postfix (האלגוריתם מובא בהמשך) לדוגמא: עבור הקלט

"(5 + 3) × ((20/10) + (8 - 6))"

הפונקציה תחזיר:

"5 3 + 20 10 / 8 6 - + *"

שים/י לב:

- הגרשיים אינם חלק מהביטוי
- הביטוי יכול להכיל מספרים בעלי יותר מספרה אחת
- הצב/י ' ' (רווח בודד) בין כל שני רכיבים בביטוי postfix המתקבל

המרת ביטוי infix (תוכי) לביטוי postfix (סופי):

אלגוריתם הקולט ביטוי בייצוג infix - תוכי, תו אחר תו וממיר אותי לביטוי בייצוג postfix – סופי הנחה: הביטוי הנקלט מכיל סוגריים עגולים בלבד (כולל סוגריים מקוננים)

1. צור/י מחרוזת ריקה $str \leftarrow$
2. צור/י מחסנית-תווים ריקה $stk \leftarrow$
3. קרא/י תו ראשון מהקלט $ch \leftarrow$
4. כל עוד לא נגמר הקלט, בצע/י:
 - 4.1 אם ch סוגר פותח - שמאלי)
 - דחוף/י את ch למחסנית
 - 4.2 אם ch סוגר סוגר - ימני)
 - שלוף/י את תוכן המחסנית עד (לא כולל) הסוגר השמאלי ושרשר/י למחרוזת str .
 - בסיום – שלוף/י את הסוגר השמאלי.
 - 4.3 אם ch אופרטור –פעולה חשבונית)
 - שלוף/י מהמחסנית את כל הפעולות בדרגת קדימויות גבוהה יותר ושרשר/י אותן למחרוזת str .
 - בסיום – דחוף/י את ch למחסנית.
 - 4.4 אם ch ספרה / מספר)
 - שרשר/י אותו למחרוזת str
 - 4.5 קרא/י את התו הבא מהקלט $ch \leftarrow$
 5. כל עוד המחסנית אינה ריקה
 - שלוף/י ושרשר/י את תוכן המחסנית למחרוזת str

ב. הגדר/י פונקציה גלובאלית בשם CalcPostfix המקבלת כפרמטר **מחרוזת** עם ביטוי חשבוני בייצוג postfix (סופי) (הביטוי יכול להכיל פעולות חיבור, חיסור, כפל וחילוק) על הפונקציה לחשב ולהחזיר את תוצאת הביטוי. (האלגוריתם מובא בהמשך)

לדוגמא עבור הקלט

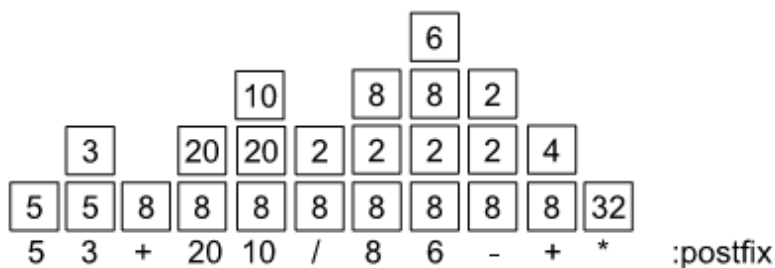
$$53 + 2010 / 86 - + *$$

הפונקציה תחזיר: 32

חישוב ביטוי postfix (סופי) באמצעות מחסנית:

אלגוריתם לחישוב ביטוי postfix – סופי

1. התחלי עם מחסנית ריקה
2. עבורי על כל ביטוי משמאל לימין:
3. אם האיבר הבא הוא אופרנד – הכנס/ אותו למחסנית
4. אם הוא פעולה – הפעלי את הפעולה על שני האיברים הנמצאים בראש המחסנית והכנס/ את התוצאה למחסנית



שים/ לב: בפונקציה infixToPostfix המחסנית מיועדת להחזקה זמנית של תווים (char) ובפונקציה CalcPostfix עבור- int. אין צורך להעתיק את מימוש המחלקה stack עבור char מכיון שהקומפילר מבצע את ההמרות מ-char ל-int ובחזרה על פי קוד ASCII.

ג. להלן תכנית ראשית הבודקת את נכונות התכנית:

```
int main()
{
    string exp;
    cout << "enter an infix expression as a string" << endl;
    cin >> exp;
    string postfix = infixToPostfix(exp);
    cout << postfix<<endl;
    cout << calcPostfix(postfix)<<endl;
    return 0;
}
```

שאלה מס' 2:

בהרצאה הוגדרה המחלקה האבסטרקטית Queue באופן הבא:

```
class Queue
{
public:
    virtual void clear() = 0;
    virtual void enqueue(int value) = 0;
    virtual int dequeue() = 0;
    virtual int front () = 0;
    virtual bool isEmpty() const = 0;
};
```

עבור מחלקה זו הוצעו שתי הצעות למימושים שונים לתור, אחת ע"י מערך (*Vector*) והשניה ע"י רשימה (*List*). בשאלה זו נדרוש מימוש נוסף (שלישי) והפעם בעזרת מחסנית (ומחסנית עזר נוספת):

```
class QueueStack : public Queue
{
protected:
    stack* data;
public:
    QueueStack();
    void clear() override;
    int dequeue() override ;
    void enqueue(int value) override;
    int front() override;
    bool isEmpty() const override;
};
```

עליך לממש מחלקה זו.

להלן תכנית ראשית הבודקת את נכונות המחלקה:

```
#include <iostream>
#include "QueueStack.h"
using namespace std;
int main() {
    Queue* Q;
    Q = new QueueStack();
    try {
        for (int i = 0; i < 10; i++)
            Q->enqueue(i);
    }
    catch (const char* msg)
    {
        cout << msg;
    }
    cout << "first on Q is: " << Q->front() << endl;
    cout << "take out 2 elemets:" << endl;
    cout << Q->dequeue() << ' ' << Q->dequeue() << endl;
    cout << "first on Q is: " << Q->front() << endl;
    Q->enqueue(8);
    Q->enqueue(9);
    while (!Q->isEmpty())
        cout << Q->dequeue() << " ";
    return 0;
}
```

בהצלחה רבה!