**סדנא ב- ++C – 181500**

**תרגיל בית מספר 8**

**Template, תור ומחסנית**

**שים/י לב:**

1. הקפד/י על קְרִיאוּת התכנית ועל עימוד (Indentation).
2. הקפד/י לבצע בדיוק את הנדרש בכל שאלה.
3. בכל אחת מהשאלות יש להגדיר פונקציות במידת הצורך עבור קריאות התכנית.
4. 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:**

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

1. הגדר/י פונקציה גלובאלית בשם המקבלת כמחרוזת ביטוי בייצוג ומחזירה מחרוזת חדשה עם ביטוי זהה בייצוג (האלגוריתם מובא בהמשך)

לדוגמא: עבור הקלט

הפונקציה תחזיר:

שים/י לב:

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

**המרת ביטוי (תוכי) לביטוי (סופי):**

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

1. צור/י מחסנית ריקה
2. צור/י מחסנית-תווים ריקה
3. קרא/י תו ראשון מהקלט
4. כל עוד לא נגמר הקלט, בצע/י:
   1. אם ( סוגר פותח - שמאלי)

דחוף/י את למחסנית

* 1. אם ( סוגר סוגר - ימני)

שלוף/י את תוכן המחסנית עד (לא כולל) הסוגר השמאלי ושרשר/י למחרוזת .

בסיום – שלוף/י את הסוגר השמאלי.

* 1. אם ( אופרטור –פעולה חשבונית)

שלוף/י מהמחסנית את כל הפעולות בדרגת קדימויות גבוהה יותר ושרשר/י אותן  
למחרוזת .  
בסיום – דחוף/י את למחסנית.

* 1. אם (ch ספרה / מספר)

שרשר/י אותו לחחרוזת

* 1. קרא/י את התו הבא מהקלט

1. כל עוד המחסנית אינה ריקה

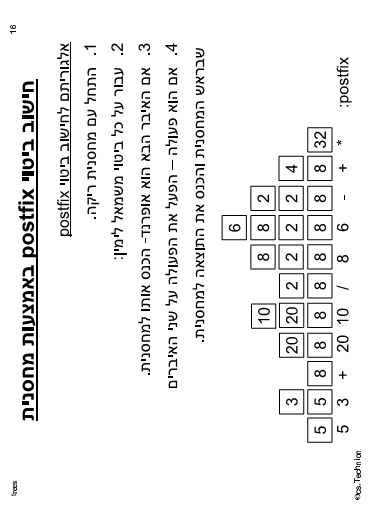
שלוף/י ושרשר/י את תוכן המחסנית למחרוזת

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

לדוגמא עבור הקלט

הפונקציה תחזיר:

**חישוב ביטוי (סופי) באמצעות מחסנית:**

אלגוריתם לחישוב ביטוי – סופי  
1. התחל/ח עם מחסנית ריקה

2. עבור/י על כל ביטוי משמאל לימין:

3. אם האיבר הבא הוא אופרנד – הכנס/י אותו למחסנית

4. אם הוא פעולה – הפעל/י את הפעולה על שני האיברים הנמצאים בראש המחסנית והכנס/י את   
 התוצאה למחסנית

שים/י לב: בפונקציה המחסנית מיועדת להחזקה זמנית של תווים () ובפונקציה עבור- . אין צורך להעתיק את מימוש המחלקה עבור מכיוון שהקומפיילר מבצע את ההמרות מ- ל- ובחזרה על פי קוד .

1. 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 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 () והשניה ע"י רשימה (). 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!**