המחלקה להנדסת מערכות תקשורת
אוני' בן-גוריון בנגב

הנחיות הגשה:

1. יש להגיש את קבצי העבודה (8 קבצים סה"כ ללא assignment1.cpp) במערכת ה VPL ולוודא שאין שגיאות קומפילציה במערכת. **אין להגיש את הקובץ assignment1.cpp , קובץ זה נמצא במעטפת ההרצה של VPL.**
2. ההגשה מתבצעת בזוגות **ועל שני בני הזוג לבצע הגשה במערכת**. בנוסף נדרש מכם להוסיף בהערות ההגשה (בדף ההגשה במודל) את השמות והת.ז של שני השותפים.
3. יש להתאים את הפלטים של התוכנית שלכם לפלטים של קובץ ההרצה לדוגמא.
4. הקפידו על הוראות ברורות וקריאות למשתמש לפני כל פעולת קלט.
5. יש להקפיד של שמות המחלקות והשיטות כפי שמצוין בעבודה, הקפידו על אותיות גדולות וקטנות (אי התאמה תגרום לשגיאת קומפילציה בבדיקה האוטומטית).
6. הקפידו על תכנות נכון, זהו חלק מן הציון:
a. • שמות משתנים בעלי משמעות.
b. • שימוש חוזר בקוד.
c. • הקפדה על הזחות.
d. • שימוש נרחב בהערות באנגלית בלבד. הקפידו לתעד כל שיטה וכל קטע קוד משמעותי.
7. בראש כל קובץ יש להוסיף בהערה את הטקסט הבא:

```
*/Assignment C++: 1
  Author: Israel Israeli, ID: 01234567
/*
```

8. עליכם להחליף את הפרטים בפרטים שלכם.
9. כל שאלה יש להפנות לפורום המתאים לשאלות במודל.
10. הארכות יינתנו רק במקרי מילואים, אבל ומחלה חריפה ובצירוף אישורים מתאימים.

תרגיל מס' 1

נושא התרגיל: מחלקה – פונקציה בונה, הורסת, public, private, העמסת אופרטורים.

<u>הערה חשובה</u>: רשימת המתודות היא **<u>חלקית</u>** בלבד.
היא מהווה את המינימום הנדרש. ייתכן ואף סביר להניח שתצטרכו להוסיף מתודות **<u>נוספות</u>** עבור המחלקות המתוארות. חישבו היטב מה באמת צריך בכל מחלקה.
לכל המחלקות חובה להוסיף Constructor ו- Destructor דיפולטיבים (אפילו אם הוא ריק). על הבנאי הדיפולטיבי לאתחל את האובייקט בערכים הגיוניים. כמו כן עליכם להוסיף מתודות set ו get-**לפי הצורך**.
בתרגיל זה אנו נבנה את המחלקות הבאות:

1. <u>טיפוס נתונים מופשט – Abstract Data Type (ADT)</u>

טיפוס נתונים מופשט (Abstract Data Type או ADT) הוא מודל מתמטי עבור קבוצה מסוימת של מבני נתונים בעלי התנהגות דומה. טיפוס נתונים מופשט מוגדר על ידי הפעולות שניתן לבצע עליו ועל ידי מגבלות שחלות על תוצאות הפעולות האלה (או העלות שלהן מבחינת סיבוכיות זמן ומקום).
לדוגמה, מחסנית היא טיפוס נתונים מופשט הכולל פעולת (Push) שמוסיפה איבר, פעולת (Pop) שמסירה את האיבר האחרון שנוסף למחסנית, ופעולת Peek המאפשרת גישה למידע בראש המחסנית ללא הסרתו. בהקשר של ניתוח סיבוכיות של אלגוריתמים העושים שימוש במחסנית, ניתן להוסיף דרישות המקילות את ניתוח הסיבוכיות: זמן קבוע לכל פעולה, ללא תלות בכמות הנתונים במחסנית, ושטח זיכרון קבוע עבור כל איבר.
ניתן לקרוא עוד בקישור הבא: https://en.wikipedia.org/wiki/Abstract_data_type.

עליכם לממש את מחסנית דינמית (מחסנית שהגודל שלה משתנה בהתאם לצורך), לשם כך ממשו את המחלקה הבאה:

<u>StackNode</u>
<u>Private members:</u>
data – int
next – StackNode*
<u>Stack</u>
<u>Private members:</u>
top – StackNode*
<u>Public members:</u>
Stack() – Initialize top to NULL.
push(int) - Inserts a new element at the top of the stack.
pop() - Removes the top element on the stack. If stack is empty, print “Stack is empty”.
isEmpty() - Returns true if the stack is empty; otherwise, it returns false
peek() - Returns the top element present in the stack without modifying the stack.
In case where the stack is empty, return INT_MIN (need to include <climits>).
The return type is integer.
<u>Operators:</u>

- + - Concatenate two ‘Stack’ objects. i.e.
  $S1 + S2$ returns $\{S1, S2\}$.
- + - Concatenate ‘Stack’ object with an integer. i.e.
  $S1 + 3 \; or \; 3 + S1$ returns $\{S1, 3\}$ or $\{3, S1\}$ respectively.
- += - push an integer to the stack. i.e.
  $S += 4$ returns $\{4, S\}$.

- == - Compare two stacks and return true if they contain the same elements in the same order, and false otherwise.
- << - Insertion operator for easy printing.

2. המחלקה myQueue:

תור (queue) הוא מבנה נתונים מופשט המוגדר על ידי הפעולות הבאות:

הכנסה - (enqueue) הוספת אובייקט אחד חדש בסופו של התור
הוצאה - (dequeue) הוצאתו של האובייקט הנמצא בראש התור
בדיקה אם התור ריק(isEmpty)
בדיקת ערך בראש התור(peek)

פעולות ההכנסה וההוצאה בתור מבוססות על העקרון נכנס ראשון - יוצא ראשון FIFO.
עליכם לממש את התור בעזרת שימוש בוקטור.
וקטור הוא מערך דינאמי במובן שגודלו יכול להשתנות לאורך התוכנית, אין חובה להקצות אותו בצורה דינאמית כמובן. לקריאה נוספת על וקטורים:
https://www.geeksforgeeks.org/vector-in-cpp-stl
http://www.cplusplus.com/reference/vector/vector

private members:
vector<int> – contain the element inside the queue.
int maxQ – the capacity of the queue.
Public members:
myQueue (int maxQ).
enQueue(int val) - insert a new element at the end of the queue. Return true if succeed and false otherwise.
deQueue() – remove the next element in the queue. Return true if succeed and false otherwise.
isEmpty() – return true if the queue is empty and false otherwise.
peek() – returns the value of the first element in the queue (return int by value). If the Queue is empty, return -1.

3. מחלקת תפריט – Menu

מחלקה זו תנהל את המערכת. על מחלקה זו לבצע פעולות קלט/פלט מהמשתמש למעט שיטות שהוגדרו לכך במפורש במחלקות אחרות .יש להציג את התפריט בלולאה עד שהמשתמש יבחר לסיים אותה.

שיטות:

mainMenu – שיטה זו תדפיס את התפריט הבא:

המשתמש מקיש	שם הפעולה	הערות
1	כניסה לתפריט מחסנית	לאחר הקשה זו המערכת תפעיל מתודה של תפריט מחסנית
2	כניסה לתפריט מחסנית	לאחר הקשה זו המערכת תפעיל מתודה של תפריט מחסנית
3	יציאה מהתוכנית	לאחר הקשה זו המערכת תדפיס Goodbye! ותצא

stackMenu – שיטה זו תדפיס למסך תפריט להפעלת המחסנית.

המשתמש מקיש	שם הפעולה	הערות
1	הוספת איבר למחסנית	הקשה זו תוסיף איבר למחסנית בהתאם לקלט שמתקבל מהמשתמש
2	הוצאת איבר מהמחסנית	הקשה זו תשלוף איבר מהמחסנית
3	בדיקה אם המחסנית ריקה	הקשה זו תדפיס הודעה מתאימה בהתאם לתכולת המחסנית
4	הדפסת איברי המחסנית	הקשה זו תציג את איברי המחסנית לפי הסדר
5	יציאה מהתוכנית	יציאה מתפריט זה וחזרה לתפריט הראשי

queueMenu() - בעת הפעלת שיטה זו התוכנית תיתן למשתמש לבחור את גודל התור. לאחר מכן המשתמש יוכל לבצע פעולות על התור.

המשתמש מקיש	שם הפעולה	הערות
1	הדפסת התור	
2	הוספת איבר לתור	
3	הסרת איבר מהתור	
4	הדפסת האיבר הראשון בתור	
5	יציאה מהתוכנית	יציאה מתפריט זה וחזרה לתפריט הראשי

ה main -

מייצרת אובייקט מסוג תפריט ומריצה את המתודה "תפריט ראשי" וזהו.

הערות:

1. הקלטים יהיו מהטיפוסים החוקיים. ז"א בכל מקום נכניס את הטיפוס המתאים. אנחנו לא מתחייבים שהוא יהיה בטווח מסוים – אלא אם כן נאמר אחרת בשאלה עצמה.
2. אחרי כל הדפסה יש לבצע ירידת שורה.

יש להקפיד על תכנות נכון:

a. כל הערכים שהם קבועים (מבחינה לוגית הם לא אמורים להשתנות), חייבים להיות מוגדרים כ: const, define או, enum בהתאם לצורך.
b. יש לרשום הערות בשפע, ובאנגלית בלבד (לכל מחלקה מה התפקיד שלה, התוכנית מה היא מבצעת, כל פעולה לא טריוויאלית להסביר, וכל 2~3 שורות קוד– הערה, כל מתודה מה היא עושה).
c. יש להקפיד על כימוס נכון – כל השדות ומתודות השירות ב-private, הממשק ב-public, חלוקה לקבצים.
d. יש להקפיד על הזחות, כיתוב נכון וקריא, ושמות משמעותיים.
e. יש לנסות ולייעל את הקוד והתוכנית ככל שניתן. הקפידו על reuse בקוד . נקודות ירדו על קוד כפול !
f. לפני בקשת קלט, יש להדפיס למשתמש הוראה (איזה קלט מבוקש)
g. להזכירכם: הנכם נדרשים לתכנת בשפת C++ אי-לכך, כל שימוש בפונקציות וספריות של שפת C אסורה.

3. עבור הצלחת הבדיקות עליכם לוודא שהדפסות זהות לקובץ ההרצה שקיבלתם. (לא יורדו נקודות על שוני ברווחים ושורות).
4. בהצלחה!