

המחלקה להנדסת מערכות תקשורת אוני' בן גוריון בנגב

הנחיות הגשה:

1. יש להגיש את קבצי העבודה (6 קבצים סה"כ ללא assignment2.cpp) במערכת ה VPL ולוודא שאין שגיאות קומפילציה במערכת. **אין להגיש את הקובץ assignment2.cpp , קובץ זה נמצא במעטפת ההרצה של VPL.**

2. ההגשה מתבצעת בזוגות, ואין צורך להגיש פעמיים עבור שני בני הזוג.
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).
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' 2

נושא התרגיל: העמסת אופרטורים ושימוש בוקטורים

בתרגיל זה אנו נבנה את המחלקות הבאות :

הערה חשובה: רשימת המתודות היא **חלקית** בלבד.

היא מהווה את המינימום הנדרש ממכם. ייתכן ואף סביר להניח שתצטרכו להוסיף מתודות נוספות עבור המחלקות המתוארות. חישבו היטב מה באמת צריך בכל מחלקה. לכל המחלקות חובה להוסיף Constructor ו- Destructor דיפולטיביים (אפילו אם הוא ריק). כמו כן עליכם להוסיף מתודות set | get לפי הצורך.

1. המחלקה Set:

כתבו את המחלקה Set אשר תחזיק נתונים לגבי קבוצה של מספרים. קבוצה היא אוסף של עצמים. כל עצם בעולם, או שהוא שייך לקבוצה (ואז הוא נקרא איבר של הקבוצה) או שאינו שייך לה. לא ניתן להיות איבר של קבוצה יותר מפעם אחת. שתי קבוצות הן שוות כאשר יש להן בדיוק אותם האיברים.

private members:

int maxSize – Maximal number of elements.

int currentSize – Current number of elements.

int* arr

Public members:

Set() – Initialize maxSize to 1000 and allocate memory for an integer array with the size maxSize.

Set(int maxSize) - Initialize maxSize and allocate memory for an integer array with the size maxSize.

getCurrentSize() – Return currentSize.

Copy constructor.

~Set() .

בנוסף יש לממש את האופרטורים הבאים:

- = - מקבל אובייקט מסוג Set ומבצע השמה לאובייקט המפעיל.
- += - מקבל מספר שלם ומוסיף אותו לקבוצה.
- + - מקבל מספר שלם ומחזיר קבוצה חדשה אשר מכילה את איברי הקבוצה ואת המספר (גודלה כמספר האיברים בקבוצה).
- + - מקבל אובייקט מסוג Set ומחזיר קבוצה חדשה שהיא איחוד שתי הקבוצות (גודלה כמספר האיברים בקבוצה).
- > - משווה בין גדלי הקבוצות. מחזיר TRUE אם בקבוצה של האובייקט המפעיל יש יותר איברים מאשר בקבוצה של הפרמטר.
- [] - מחזיר את האיבר במיקום i בקבוצה. ניתן להניח ש i – מהווה אינדקס בגבולות הקבוצה. האופרטור אינו תומך בשינוי האיבר.
- << - הדפסה

יש לחשוב מה מטרת האופרטורים, כיצד הם משמשים אותנו עבור טיפוסים מוכרים ועל ידי כך לחכות את הפעולות שלהם.

נדרש לממש את האופרטורים כך שתתאפשר סימטריה בקוד, למשל עבור הסט s1 אשר מייצג את הקבוצה {1,3} נרצה לאפשר חיבור משני הצדדים

s1 + 4;

4 + s1;

עבור שתי הפקודות נצפה לקבל קבוצה חדשה {1,3,4}.

בנוסף נדרש להקפיד על שימוש מתאים ב const בהתאם למה לנלמד בקורס (כלל אצבע – נגדיר const היכן שאפשר להגדיר).

הערה: במצב שבו מנסים להוסיף איבר לקבוצה שבה currentSize == maxSize

אזי יש להקצות זיכרון חדש שיוכל להכיל את הקבוצה הקודמת ביחד עם האיבר החדש.

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.
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() – Return a pointer to the first element in the queue. If the Queue is empty, return nullptr.

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 של סטים	לאחר הקשה זו המערכת תפעיל מתודה של תפריט סט
2	כניסה לתפריט תור	לאחר הקשה זו המערכת תפעיל מתודה של תפריט תור
3	יציאה מהתוכנית	לאחר הקשה זו המערכת תדפיס Goodbye! ותצא

setMenu() – בעת הפעלת שיטה זו התוכנית תיצור שני סטים (דיפולטיבים) ותאפשר למשתמש לבצע עליהם פעולות.

המשתמש מקיש	שם הפעולה	הערות
1	הדפסת הסט הראשון	הפעולות התבצעו ע"י שימוש באופרטורים שהוגדרו.
2	הדפסת הסט השני	
3	הוספת אלמנט לסט הראשון	
4	הוספת אלמנט לסט השני	
5	הצגת האיחוד של הסטים	
6	יציאה	יציאה מתפריט זה וחזרה לתפריט הראשי

queueMenu() – בעת הפעלת שיטה זו התוכנית 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: `define`, `const`, 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. להזכירם: 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!