

בעבודה זו אתם נדרשים לתכנן, לממש ולהדגים מחלקת תור עדיפויות, Priority Queue. תור עדיפויות הוא תור בו לכל אלמנט יש שני ערכים לפחות:

1. Value

2. priority

בתור עדיפויות אלמנט עם עדיפות גבוהה מקבל שירות לפני אלמנט עם עדיפות נמוכה. אם לשני אלמנטים יש אותה עדיפות, הם מקבלים שירות לפי הסדר שלהם בתור.

שימו לב:

- התור ימומש כמערך שיכול לגדול כך שניתן להוסיף אלמנטים ללא הגבלה.
- עליכם לתכנן את התור באמצעות תבנית (template).
- אסור להשתמש במבני הנתונים מהספריה STL.

המתודות שחובה לממש:

קבלת האלמנט בתחילת התור (בעל העדיפות הגבוהה ביותר)	top
חיווי אם התור ריק	empty
מספר האלמנטים בתור	size
הוספת אלמנט לתור	push
הוצאת האלמנט שבראש התור	pop

עליכם לצרף לעבודה:

- ציור המתאר את שלבי ההוספה וההסרה

- הסיבוכיות שקיבלתם עבור כל אחת מהמתודות

כמו כן יש לממש פונקציה חיצונית למחלקה בשם print_reversed_queue אשר מקבלת את התור ומדפיסה אותו מהסוף להתחלה. שימו לב להעביר את התור כ const reference.

מצורף טסט למחלקה PriorityQueue ובו נוצר תור של ints, תור של strings ותור של Array. כמו כן נוצר Array של PriorityQueue.
Array זו המחלקה מהמעבדה האחרונה (שימו לב שגם היא class template). ניתן להשתמש בפתרון הרשמי למעבדה שהעליתי.

```
void test()
{
    PriorityQueue<int> pq_int;
    for (int i = 0; i < 5; i++)
        pq_int.push(i);
    cout << "top = " << pq_int.top() << endl;
    cout << "int queue: " << endl;
    cout << pq_int;
    cout << "and to make sure you didn't change the queue: " << endl;
    cout << pq_int;

    PriorityQueue<string> pq_str;
    pq_str.push("This");
    pq_str.push("is");
    pq_str.push("project");
    pq_str.push("3");
    cout << "top = " << pq_str.top() << endl;
    cout << "str queue: " << endl;
    cout << pq_str;

    Array<double> da1(5);
    for (int i = 0; i < 5; i++)
        da1[i] = i;
    Array<double> da2(2);
    for (int i = 0; i < 2; i++)
        da2[i] = i;
    PriorityQueue<Array<double>> pq_array;
    pq_array.push(da1);
    pq_array.push(da2);
    cout << "Array queue reversed: " << endl;
    print_reversed_queue(pq_array);

    Array<PriorityQueue<int>> arr_pq_int;
    arr_pq_int[0] = pq_int;
    arr_pq_int[1] = pq_int;
    cout << "Array of priority queues of ints: " << endl;
    cout << arr_pq_int;
}
```

אופן ההגשה:

1. הגשה בזוגות.
2. העבודה תוגש מודפסת על עמודים בגודל A4 ומשודכת כך שתכיל:

a. בעמוד הראשון כותרת העבודה בפורמט הבא:

עבודה ב OOP 31631 סמסטר א' תשע"ה, עבודה 3, תור עדיפויות
תאריך ההגשה: dd/mm/yyyy
שם הסטודנט, תעודת זהות, מייל: שם משפחה, שם פרטי, ID, Email
שם הסטודנט, תעודת זהות, מייל: שם משפחה, שם פרטי, ID, Email
ציון:

b. בהמשך העמוד הראשון

תחת הכותרת "תאור התוכנית" תאור מילולי טכני ותמציתי של מחלקות התוכנית (Data members ו Function members) והשיטה/האלגוריתם שבה נכתבה התוכנית.

התאור צריך להיות כזה שאם הוא ינתן למתכנת אחר ביחד עם הגדרת הבעיה המתכנת יכתוב תוכנית שפועלת באותה השיטה כמו התוכנית שלכם.

c. תחת הכותרת "מקורות" רשימה של כל המקורות בהם נעזרתם בכתיבת התוכנית ומה נלקח מכל מקור.

d. תחת הכותרת "בעיות ידועות" רשימה של הבעיות הידועות בתוכנית (אם קיימות)

e. בעמוד חדש תדפיס של קבצי התוכנית. ההגשה תכלול את הקבצים הבאים:

Array.h

PriorityQueue.h

f. על התוכנית להיות כתובה באופן מסודר ומדורג (הזזה ימינה עם פתיחה של כל סוגריים מסולסים), יש לתת שמות משתנים משמעותיים, יש להוסיף הערות הסבר באנגלית בתוכנית, המנעו משכפול קוד ומפונקציות ארוכות, העברת אובייקטים תעשה by reference ובמידת האפשר const, מתודות שיכולות להיות const יש להכריז עליהן ככאלו.

g. בעמוד חדש תדפיסו בדיקת גלישת זכרון.

3. בנוסף להדפסות יש להגיש את כל קבצי הקוד שכתבתם למטלת ההגשה במודל:

a. על הקבצים להיות מכווצים בתוכנת zip ולא בשום תוכנת כיווץ אחרת.

b. שם הקובץ zip יהיה ה.ת.ז. של הסטודנטים כך: id1_id2_hw2.zip.

c. ההגשה תכלול את הקבצים הבאים:

Array.h

PriorityQueue.h

צוות הקורס שומר לעצמו את הזכות לזמן את הסטודנט/ים לבחינה בעל פה על העבודה שהוגשה.

בהצלחה!