



המכללה האקדמית להנדסה סמי שמעון

עבודת הגשה מס' 2

(הכלה, אופרטורים, friend)

מועד הגשה עד: 15.08.2022 שעה 23:55הוראות הגשה:

- (1) **אנא קראו בעיון את כל תיאור העבודה בטרם תתחילו לכתוב קוד.**
- (2) הגשה באופן עצמאי בלבד. הגשה בקבוצות תוביל לציון 0 בעבודה.
- (3) אין לשתף או להעתיק את העבודה או חלקים ממנה. עבירה על הוראה זו תוביל לציון 0 בעבודה.
- (4) הגשה דרך מערכת מודול בלבד. **שום עבודה לא מתקבלת במייל!**
- (5) יש למקם כל מחלקה שיהיה עליכם ליצור, בשני קבצים נפרדים H ו-CPP. יש להכניס את החלק התיאורטי בקובץ וורד נפרד. יש להכניס את כל הקבצים של החלק המעשי + קובץ הוורד לתיקיה אחת, ואז לכווץ יחד. נדרש להגיש קובץ אחד בפורמט RAR או ZIP המכיל את כל הקבצים של כל השאלות. לקובץ המכווץ יהיה שם המהווה את מספר ת.ז. של המגיש.

חלק תאורטי

- (1) **האם ניתן להשתמש באופרטור += בעל שני פרמטרים מסוגים שונים? במידה ולא יש לתת הסבר, אם כן יש לתת דוגמא.**
- (2) **האם הטענה נכונה או לא נכונה: ניתן לתת הוראה static רק למשתנים של מחלקות. יש לנמק.**
- (3) **מה היא המוטיבציה בהגדרת friend בין מחלקות?**
- (4) **בתוכנית שבה יש מספר מחלקות, החליט המתכנת לבדוק את כמות הפעמים שמשתמשים במטודה של אחד המחלקות.**
- תאר ותן דוגמא מה צריך להגדיר מתכנת.**
- (5) **האם קיים אופרטור שניתן להשתמש בו באותה מחלקה גם עם פרמטר וגם בלי פרמטר, אם כן יש לתת דוגמא.**
- (6) **רשום מהו הבדל בין שלושה הגדרות המשתנים שמשתמשות ב-const.**

```
int x[] = { 5,4,3,2,1 };
const int * ptr1 = x;
int * const ptr2 = x;
const int * const ptr3 = x;
```

- (7) **נתון קטע של קוד שעובר קומפילציה אך נותן טעות בעת יצירת אובייקט. איך ניתן לתקן את הפונקציה בנאי ולהמשיך להשתמש באופרטור המוגדר.**

```
class A{
private:
    int * num;
    int snum;
public:
    A(int n) :snum(n) {
        num = new int[n];
        for (int i = 0; i < snum; i++)
            this[i] = i + 1;
    }
    int& operator[](int x) {
        return num[x];
    }
    . . . . .
};
```



חלק מעשי

המערכת:

בתרגיל הזה אתם מתבקשים לבנות את מערכת שמצפינה טקסט(הצפנה) או משחזרת טקסט מוצפן(פענוח). הצפנה או פענוח מתבצע ע"י מפתח הצפנה המכיל את פעולות שצריך לבצע על טקסט. הפעולות מתבצעות בשתי רמות: רמת מילה או ברמת כל טקסט.

לצורך ביצוע המשימה אתם מתבקשים לבנות את המחלקות הבאות:

- (1) מחלקה **Word** המתארת מילה (מערך דינאמי של תווים מסוג **char** וגודל מערך). יש להגדיר בה את כל הבנאים הדרושים, הורס ואופרטורים הבאים:
 - (א) ! - שמחליף תווים לפי כללים הבאים: אם תו הינו "ספרה" יש להחליף לספרה משלימה ל-9 (לדוגמא: 0<-9, 3<-6, 7<-2), אם תו הוא "אות" יש להחליף אותיות קטנות לגדולות והפוך וגם לאות משלימה. (לדוגמא: a<-Z, Y<-b, d<-W, K<-p)
 - (ב) & - הופך סדר תווים במילה. (לדוגמא: 'abc23' <- '32cba')
 - (ג) << - סיבוב מעגלי שמאלה ל-n מקומות. (לדוגמא ל-2 מקומות: 'abc23' <- 'c23ab')
 - (ד) >> - סיבוב מעגלי ימינה ל-n מקומות. (לדוגמא ל-2 מקומות: 'abc23' <- '23abc')
 - (ה) += - להוספת תו אקראי במקום n במילה. (לדוגמא מקום 2: 'abc23' <- 'abTc23')
 - (ו) -= - להורדת תו במקום n במילה. (לדוגמא מקום 2: 'abc23' <- 'abTc23')
 - (ז) [] - המחזיר תו (יש לממש עם אופציית עדכון).
 - (ח) << cout - מוסיף מילה לזרימת פלט.

- (2) מחלקה **Text** המתארת את טקסט כמערך דינאמי של מצבעים לאובייקטים מסוג מילה(**Word**) וכמות מילים בטקסט(**מספר שלם**). יש להגדיר בה את כל הבנאים הדרושים, הורס ואופרטורים:
 - (א) ! - שמחליף תווים לכל המילים בטקסט ע"י הפעלת אופרטור מתאים לכל מילה. ("abc 123"<-"ZYX 876")
 - (ב) & - שהופך סדר מילים בטקסט. ("aaa bbb ccc"<-"ccc bbb aaa")
 - (ג) << - סיבוב מעגלי מילים שמאלה ל-n מקומות.
 - (ד) >> - סיבוב מעגלי מילים ימינה ל-n מקומות.
 - (ה) += - להוספת מילה בגודל אקראי במקום n בטקסט.
 - (ו) -= - להורדת מילה במקום n בטקסט.
 - (ז) [] - המחזיר מילה (יש לממש עם אופציית עדכון).
 - (ח) << cout - מוסיף טקסט לזרימת פלט.



3) מחלקה **EncrypteText** המתארת אובייקט שמורכב מתכונות:

- מצביע לאובייקט **Text**.
- סטטוס (טקסט להצפנה 0 או פענוח 1).
- מפתח הצפנה/פענוח: מערך דינאמי חד-ממדי של מספרים שלמים המורכב משלישיות מספרים:
 - קוד פעולה (אחד מהאופרטורים הבאים: 1 - !, 2 - &, 3 - <<, 4 - >>, 5 - +=)
 - ביצוע הפעולה: על כל הטקסט (-1), על מילה מסוימת (0)
 - פרמטר עזר לפעולה המתבצעת. לדוגמא בפעולת הזזה, אם ערך של פרמטר הוא 3, אז תבצע הזזה לשלוש מקומות.
 - במקרה שיש הזזה במילה הפרמטר הוא מספר תלת ספרתי: שני ספרות ראשונות מספר מילה מתחילת טקסט וספרה אחרונה מספר תווים להזזה. לדוגמא 123(12) מסי מילה ו-3 תווים להזזה).
- א) בנאי שמקבל טקסט להצפנה או פענוח וסטטוס.
- ב) פונקציה הורסת.
- ג) אופרטור += - שמקבל מפתח, מערך מספרים (ניתן לממש כך שאחד ממספרים יסמן את סוף המערך לדוגמא 2-).
- ד) אופרטור ! - שמבצע הצפנת טקסט.
- ה) אופרטור & - שמבצע פענוח של הטקסט.
- ו) אופרטור << cout - שמוסיף לזרימת פלט טקסט מוצפן ומפתח או טקסט לאחר פענוח.

*תהליך פענוח מתבצע בסדר הפוך לתהליך הצפנה
*אין צורך בבדיקת תקינות קלט, יש להניח שכל הקלט שנקלט הינו חוקי ותקין.

4) בתוכנית הראשית (main) יש לתת:

שתי דוגמאות: דוגמה להצפנת טקסט ע"י הוספת מפתח הצפנה ודוגמה ליצירת אובייקט **EncryptedText** עם טקסט לפענוח, הוספת קוד לפענוח, פענוח והדפסת תוצאה - טקסט אחרי פענוח.

5) הערות:

- א) אין גישה ישירה למשתני המחלקה מבחוץ.
- ב) ניתן להוסיף פונקציות נוספות לפי הצורך. עליכם לבחור היטב את ההרשאה המתאימה לכל פונקציה.
- ג) אין להוסיף תכונות (משתני מחלקות).
- ד) חלה חובה על שחרור כל הזיכרון הדינאמי שהוקצה בתוכנית!
- ה) תהליך הצפנה ופענוח יש לבצע באמצעות אופרטורים **בלבד!**
- ו) קלט תקין (טקסט) יכול להיות מורכב מ-: תווים שהם אותיות גדולות או קטנות וספרות. רווח או רווחים מפרידים בין המילים. ניתן להוסיף תו לסימון סוף הטקסט (לדוגמא .).
- ז) בתרגיל מותר להשתמש בספריות `iostream`, `assert`, `time.h` **בלבד!**
- ח) כל הערכים שהם קבועים, (מבחינה לוגית הם לא אמורים להשתנות), חייבים להיות מוגדרים כ: `const`, `define` או `enum`, בהתאם לצורך.
- ט) יש לרשום הערות לכל מחלקה ופונקציה מה התפקיד שלה, כל תוכנית מה היא מבצעת, כל פעולה לא טריוויאלית להסביר, וכל 2~3 שורות קוד – הערה, כל מתודה מה היא עושה)
- י) יש להקפיד על כימוס נכון - כל השדות ומתודות השירות ב-**private** והממשק ב-**public!**
כמו כן: חלוקה לקבצים! כל מחלקה בקבצים נפרדים!



(יא) יש להקפיד על הזחות!!! כיתוב נכון וקריא! ושמות משמעותיים.
(יב) יש לנסות ולייעל את הקוד והתוכנית ככל שניתן. הקפידו על **reuse** בקוד.

דוגמא

נניח שהטקסט שנכנס להצפנה היה:

The London is capital of Great Britain.

והקוד הצפנה היה :

{3,-1,4, 1,-1,0, 1,0,4, 4,-1,2, 5,-1,2, -2}

- הקוד הראשון שמופעל:

3,-1,4

הטקסט לאחר ההצפנה:

of Great Britain The London is capital.

- הקוד השני שמופעל:

1,-1,0

הטקסט לאחר ההצפנה:

LU tIVZG yIRGZRM gSV oLMWLM RH XZKRGZO.

- הקוד השלישי שמופעל:

1,0,4

הטקסט לאחר ההצפנה:

LU tIVZG yIRGRZM gSV London RH XZKRGZO.

- הקוד הרביעי שמופעל:

4,-1,2

הטקסט לאחר ההצפנה:

RH XZKRGZO LU tIVZG yIRGZRM gSV London.

- הקוד החמישי שמופעל:

5,-1,2

הטקסט לאחר ההצפנה:

RH XZKRGZO T4I5y LU tIVZG yIRGZRM gSV London.

בהצלחה!