

LACPP

תרגיל ראשון - אובייקטים
הגשה: יום רביעי ה - 15.12.04

1.

א. (7%) התבוננו בקובץ q1A.cpp שבתיקית התרגיל. הסבירו מה קורה בזמן ההרצה שלו ומדוע הוא פגום.

ב. (15%) התבוננו בקובץ q2B.cpp והסבירו את אופן פעולתו. תארו כמה זיכרון מוקצה על המחשנית וכמה על הערמה, מתי מוקצה כל סוג של זיכרון ומתי הוא משוחרר.

2. (38%)

כתבו קוד של אובייקט המסוגל לבדוק את השערת גולדבך עבור כל המספרים הטבעיים עד מספר נתון.

השערת גולדבך טוענת שכל מספר זוגי גדול מ - 2 אפשר להציג כסכום של שני מספרים ראשוניים.

ה - class שתכתבו ייקרא Goldbach. אובייקט מסוג Goldbach אמור לענות ב - $O(1)$ על השאלה אם מספר הנמצא בטווח הוא מספר ראשוני, להדפיס את כל המספרים הראשוניים בטווח ולהדפיס את כל הפרוקים של המספרים הזוגיים הגדולים משתיים. לאובייקט תהיה פונקציה נוספת שתאפשר את שינוי הטווח. פונקציה זו יכולה לעשות גם עבודת הכנה פנימית. בתיקית האתר נמצא הקובץ q2.cpp שישמש כקובץ ה - main שלכם. q2.out מגדיר איך אמור להיות הפלט שלכם.

הוכחת השערת גולדבך תחשב כבונוס (עד $1e6$ נקודות).

3. (40%)

בשאלה זו אתם מתבקשים לכתוב רשימה משורשרת דו כיוונית שקדקודיה אינם מכילים מצביעים.

קדקוד ברשימה דו כיוונית רגילה מכיל שני מצביעים ואת הנתונים שהוא מאכסן. ברשימה שתבנו, קודקוד ידרוש פחות זיכרון:

```
class Node {  
    // the only data members:  
    int _link;  
    double _data;  
    // ... member functions ...  
};
```

כפי שניתן לראות, במקום שני מצביעים, לקדקוד זה יש משתנה בודד מסוג int הנקרא link. הערך של link יהיה פונקציה של הכתובות של הקדקוד שלפניו והקדקוד שאחריו. הדרישה מהפונקציה היא שניתן יהיה לשחזר את כתובת אחד הקדקודים מתוך ערך הפונקציה וכתובת הקודקוד השני. אם כתובת הקדקוד שלפני, היא p1, כתובת הקדקוד שאחרי, היא p2 והפונקציה היא פונקציית החיבור, אז השחזור יהיה באמצעות פעולת חיסור:

```
link = p1 + p2;  
p2 = link - p1;
```

(יש לדאוג כמובן שהחישובים יעשו ב - int-ים)

ערך פעולת החיבור עלול להיות גדול מה- `int` המקסימאלי ואז תהיה גלישה. אם מעוניינים להימנע מהתייחסות לבעיית גלישה, ניתן להשתמש בפונקציית `xor` בביטים:

```
link = p1 ^ p2;  
p2 = link ^ p1;
```

(גם כאן הפעולות חייבות להיות ב- `int`-ים)

ברגע שכל קדקוד ברשימה מאפשר לדעת את הקדקוד הבא אחריו בהינתן זה שלפניו, ניתן לרוץ על הרשימה בכיוון אחד. כאשר קדקוד מאפשר לדעת את כתובת הקדקוד שלפניו על פי הקדקוד שאחריו, ניתן לרוץ על הרשימה בכיוון ההפוך.

שימו לב שהמימוש המתואר מניח ש: `sizeof(void *) >= sizeof(int)`. זה אכן המקרה בפלטפורמה שלכם. בפלטפורמה אחרת ייתכן שיהיה צורך לבצע שינוי.

הקובץ `q3.cpp` אמור לשמש כקובץ ה- `main` שלכם והקובץ `q3.out` מגדיר את הפלט המבוקש.

הגשה:

קבצי קוד, קבצי `readme`, `header` ו- `makefile` המייצר בפקודת `make` בודדת את קבצי ההרצה הבאים:

`q2` המורכב מ- `q2.cpp` הנתון והקוד שלכם.

`q3` המורכב מ- `q3.cpp` הנתון והקוד שלכם.

(אין צורך להוסיף את הקבצים `q1A.cpp` ו- `q1B.cpp`)

את תדפיסי הקבצים והתשובות לשאלה 1 יש להגיש לתא הקורס.

הערה: אין דרישה להתאמת `diff` מדויקת לחלוטין. אם ההבדל בין הפלט שלכם לקובצי הפלט הנתונים הוא רק רווחים או הבדלים לא נראים אחרים, אין צורך לשנות את התכנית שלכם.

בהצלחה !