תרגיל 5, מבוא לתכנות מערכות, חורף 2020-2021

הגשה **בזוגות או ביחידים** דרך המודל
התרגיל הוא להגשה עד ליום שני, 13/1/2020 בשעה 23:59

תיאור התרגיל

בתרגיל זה תממשו מודול שמאפשר לשמור מספרים שלמים אי-שליליים בכל גודל שנבחר ולבצע עליו פעולות. המנעו ככל האפשר משכפול קוד ותחסכו לעצמכם זמן ונקודות יקרות!

מבנה התוכנית

התוכנית מורכבת ממודול בודד בשם Biggie. קובץ ה-header של Biggie (Biggie.h) וכן הבסיס של קובץ המימוש (Biggie.c) מצורפים לתרגיל.
עליכם לממש את הפונקציות הבאות שחתימתן נתונה בקובץ Biggie.h:

1. BiggieCreate – מקבלת את מספר הבתים אותו נרצה להקצות לשמירת מספרים באובייקט. יוצרת אובייקט Biggie בהתאם אשר מכיל את המספר 0.
2. BiggieCreateFromBiggie – מקבלת אלמנט מטיפוס Biggie ויוצרת אלמנט חדש שזהה לו.
3. BiggieCreateFromUInt – מקבלת מספר מטיפוס unsigned int ויוצרת Biggie שמכיל את אותו המספר עם מספר בתים זהה לזה של unsigned int.
4. BiggieDestroy – destructor ל-Biggie.
5. BiggieResize – מקבלת Biggie וכן גודל חדש עבורו ומשנה את גודל הזכרון שמוקצה לשמירת המספר. במידה והגודל החדש קטן מהישן תמחק הפונקציה את החלק ששומר את ה-most significant bits.
6. BiggieLeftShift1 – מזיזה את הביטים ב-Biggie צעד אחד שמאלה.
7. BiggieLeftShift – מקבלת Biggie ומספר n, מזיזה את המספר השמור ב-Biggie ב-n ביטים שמאלה.
8. BiggieRightShift1 – מזיזה את הביטים ב-Biggie צעד אחד ימינה
9. BiggieRightShift – כמו LeftShift רק ימינה.
10. BiggieXor – מקבלת שני Biggie (bn1 ו-bn2), מבצעת פעולת xor ביניהם, מחזירה את התוצאה ב-Biggie חדש. אם אחד ה-Biggie ארוך מהשני אז התייחסו לביטים ה"חסרים" במספר הקצר יותר כאל 0.
11. BiggieAnd – כמו BiggieXor רק עם פעולת BiggieAnd.
12. BiggieOr – כמו BiggieXor רק עם פעולת BiggieOr.
13. BiggieNot – מקבלת Biggie, מחזירה Biggie חדש שהוא תוצאת הפעלת פעולת not על ה-Biggie המקורי.
14. BiggieCopy – מקבלת שני Biggie (bn1 ו-bn2), מעתיקה את התוכן של bn2 ל-bn1. מגדילה את הזכרון של bn1 אם צריך.
15. BiggieConvert – מקבלת Biggie, מחזירה unsigned int שמכיל את המספר אותו מכיל Biggie. במידה וה-Biggie ארוך מדי אז הערך המוחזר יכיל רק את ה-(sizeof(unsigned int בתים ה-least significant.
16. BiggieNumBits – מחזירה את אורך המספר השמור ב-Biggie בביטים. למשל: עבור 4 תחזיר הפונקציה 3 (4=100b)
17. BiggieGT – מקבלת שני Biggie, מחזירה true אם הראשון גדול מהשני או false אחרת.
18. BiggieLT – מקבלת שני Biggie, מחזירה true אם הראשון קטן מהשני או false אחרת.
19. BiggieEQ – מקבלת שני Biggie, מחזירה true אם שניהם שווים.

20. BiggieAdd – מקבלת שני Biggie, מחזירה Biggie שמכיל את הסכום של שניהם. לצורך ביצוע פעולת החיבור ממשו את האלגוריתם הבא:

```
INPUT: x, y
OUTPUT: x+y
z = max(x, y)
z = resize(z, sizeof(z)+1) # Add a byte with 0 to z
w = min(x, y)
w = resize(w, sizeof(w)+1) # Add a byte with 0 to w

carry = 0
FOR i in 0:(num_bytes_in(w)-1):
   # The following is byte addition, which means that if the results
   # is 256 (0x100) then z[i] will be 0x00.
   z[i] = z[i] + carry
   IF z[i] == 0 and carry == 1 THEN
      carry = 1
   ELSE
      carry = 0
   FI

   # The following is an unsigned int addition which means that if
   # the result is bigger than a byte we still want to keep all the bits
   IF z[i]+w[i] > 0xFF THEN:
     carry = 1
   FI
   z[i] = z[i] + w[i]
DONE
```

21. BiggieMultiply – מחזירה Biggie שמכיל את מכפלת שני ה-Biggie שהתקבלו כקלט. אפשר להשתמש באלגוריתם המתואר כאן:
http://www.geeksforgeeks.org/russian-peasant-multiply-two-numbers-using-bitwise-operators/

22. BiggieCreateFromString – מקבלת מחרוזת שמכילה מספר ויוצרת Biggie זהה למספר שהועבר. גודל ה-Biggie שנוצר צריך להיות בהתאם לגודל המספר. למשל: אם אורך המספר הוא 66 ביטים אז גודל ה-Biggie צריך להיות 9 בתים.
רמז: חישבו כיצד ניתן לממש פונקציה זו בעזרת פונקציות אחרות. אלגוריתם אפשרי יכול להתבסס על לולאה שבה בכל איטרציה תשלפו את הספרה הבאה במספר, תכפילו המספר שכבר יצרתם ב-10 ותוסיפו את הספרה החדשה.

דגשים

- הקפידו על תכנות נכון! דאגו לקוד מסודר ולהערות היכן שצריך (ורק איפה שצריך).
- התרגיל נכתב מתוך כוונה שמי שלא ישתמש בקוד של פונקציות אחרות יעבוד קשה מאוד (ויפסיד הרבה נקודות).
- בדקו את התוכנית בעזרת קובץ ה-main המצורף. אתם מוזמנים להוסיף לו בדיקות משלכם על מנת לוודא שהפונקציות שכתבם נכונות.
- במקרים של שימוש בזיכרון דינאמי, יש לוודא כי ההקצאות אכן ניתנו ע"י מערכת ההפעלה, וכן יש לנהל בקפידה את הזיכרון ולדאוג שבסיום השימוש כל זיכרון דינאמי שהוקצה גם ישוחרר.

- יש לוודא כי התכנית עוברת קומפילציית **gcc** על שרת החוג ללא כל שגיאות או אזהרות כלשהן, ורצה בהצלחה.

הגשה

- **הגשה אלקטרונית**: יש להגיש במערכת Moodle את הקובץ Biggie.c בלבד כמו שהוא (לא ב-zip או tar). כתבו את השמות+מספרי ת.ז. של המגישים בהערה בתחילת הקובץ.

בהצלחה!