

תכנות מערכות א' מטלה 1

קלט, פלט, ספריות Makefile

לפניות ליבגני בנושאי הקורס והמטלה:

software.systems.a.ariel@gmail.com

שימו לב:

- המטלה היא בזוגות או יחידים – לא שלשות.
- על איחור עד 3 ימים ירדו 20 נקודות. מעבר לכך המטלה לא תתקבל
- עליכם לבצע את פקודת הקומפילציה עם הדגל -Wall על מנת לוודא שתוכניתכם מתקמפלת ללא אזהרות. תכנית שמתקמפלת עם אזהרות תגרוור הורדת נקודות.
- עליכם לוודא שתוכניתכם מתקמפלת ורצה על גבי מערכת הפעלה Ubuntu עם קומפיילר GCC.
- יש להגיש את המטלה ב.git
- יש להגיש קובץ txt. השורה הראשונה תכלול את הכתובת של הגיט (לא URL מאתר github) שורה שניה תכלול את המזהה הcommit הרלוונטי ושורה שלישי את תעודת הזהות של הסטודנטים המגישים מופרדים ברווח.
- הנכם נדרשים לקוד קריא ונקי.
- בכל מקום בו יש צורך בשימוש בקבועים בעלי משמעות יש להגדיר אותם באמצעות define

ספריית NumClass

עליכם לכתוב ספרייה שחושפת למשתמש מספר פונקציות לסיווג מספרים. הפונקציות יוגדרו בקובץ NumClass.h כאשר מחזירים int עבור שאילתה בוליאנית:

0 is false

1 is true

על הקובץ להכיל את החתימות:

```
/* will return if a number is Armstrong number
```

An Armstrong number is an n-digit number that is equal to the sum of the nth powers of its digits.

For Example: $407 = 4^3 + 0^3 + 7^3 = 64 + 0 + 343 = 407$

```
*/
```

```
int isArmstrong(int)
```

```
/* will return if a number is a palindrome */
```

```
int isPalindrome(int)
```

```
/* will return if a number is prime*/
```

```
int isPrime(int)
```

/*Strong number is a special number whose sum of the factorial of digits is equal to the original number. For Example: 145 is strong number

$1! + 4! + 5! = 145$

*/

int isStrong(int)

את מימושי הפונקציות יש לכתוב 3 קבצים שונים.

1. קובץ ראשון basicClassification.c שיכיל את מימושי הפונקציות isPrime, isStrong

2. advancedClassificationLoop.c שיכיל את מימושי פונקציות isArmstrong, isPalindrome אשר ימומשו בעזרת לולאות

3. advancedClassificationRecursion.c שיכיל את מימושי פונקציות isArmstrong, isPalindrome אשר ימומשו בעזרת רקורסיה

תוכנית ראשית

עליכם לכתוב תוכנית שתקבל מהמשתמש 2 מספרים שלמים חיוביים עליכם להדפיס למסך 4 שורות אשר יכילו את כל המספרים הראשונים, מספרי ארמסטרונג, מספרים חזקים והפלנדרומים אשר בין שני המספרים שהתקבלו (כולל המספרים עצמם).

אין להדפיס כל פלט נוסף פרט למה שהתבקשת. עליכם להיות תואמים לחלוטין לדוגמאות הקלט והפלט שתקבלו.

Makefile

עליכם להגיש קובץ בשם makefile עם הפקודות הבאות:

- **make loops** על הקומפיילר ליצור את הספרייה הסטטית libclassloops.a שתכיל את מימושי כל הפונקציות שלכם עם המימוש בעזרת הלולאות (כולל basic)
- **make recursives** על הקומפיילר ליצור את הספרייה הסטטית libclassrec.a שתכיל את מימושי כל הפונקציות שלכם עם המימוש בעזרת רקורסיה (כולל basic)
- **make recursived** על הקומפיילר ליצור את הספרייה דינאמית libclassrec.so שתכיל את מימושי כל הפונקציות שלכם עם המימוש בעזרת רקורסיה (כולל basic)
- **make loopd** על הקומפיילר ליצור את הספרייה דינאמית libclassloops.so שתכיל את מימושי כל הפונקציות שלכם עם המימוש בעזרת לולאות (כולל basic)
- **make mains** על הקומפיילר ליצור את התוכנית הראשית שלכם בשם mains כשהיא מלונקג'ת לספרייה הסטטית הרקורסיבית. אם הספרייה כבר הייתה קיימת אין לקמפל אותה שוב.
- **make maindloop** על הקומפיילר ליצור את התוכנית הראשית שלכם בשם maindloop כשהיא מלונקג'ת לספרייה הדינאמית עם הלולאות.
- **make maindrec** על הקומפיילר ליצור את התוכנית הראשית שלכם בשם maindrec כשהיא מלונקג'ת לספרייה הדינאמית עם הרקורסיות.
- **make all** יקמפל את כל הספריות והתוכניות שלכם. אם משהו כבר קיים אין לקמפל אותו שוב.
- **make clean** ינקה את התיקה מכל הקבצים המקומפלים וישאיר רק קבצי h.c.txt. והעמל makefile

הגשה

במודל יש להגיש קובץ txt

הקובץ יכול 3 שורות. קישור לגיט שלכם (לא URL), מספר commit, תעודות זהות המגשים מופרדים ברווח.

http://github.com/evgeny/sw_systems_hw1.git

34cfa4b

123456789 987654321

על הגיט שלכם לכלול את הקבצים הבאים ברמה הראשונה:

- makefile
- NumClass.h
- basicClassification.c
- advancedClassificationLoop.c
- advancedClassificationRecursion.c
- main.c

יש לוודא שהתוכנית מתקמפלת ורצה על גבי ubuntu עם gcc. אתם רשאים לעבוד עם כל עורך קוד שנראה לכם לנכון אך סביבת הבדיקה תהיה ubuntu.

בדיקות בסיסיות מקומיות

מסופק לכם סקריפט בדיקה לתאימות הפלט שלכם עם הפלט הרצוי וכך גם מבנה הגיט. הבדיקות המסופקות הינן חלקיות לכן מעבר שלהן אינו מבטיח 100%. מצופה מכם לכתוב בדיקות נוספות בעצמכם.

להרצת הבדיקות יש לפתוח תיקיה חדשה ולהכניס לתוכנה את הסקריפט, התיקיות עם דוגמאות הקלט\פלט וקובץ ההגשה שלכם. פרט לקבצים אלו על התיקיה להיות ריקה. כדי להריץ את הבדיקה יש להיכנס לתיקיה שיצרתם לכתוב בשורת הפקודה:

```
python check_submission.py<your_submission_file>
```

בהצלחה!