

תרגיל בית 5 - C++ מסכם

מועד ההגשה:	יום ג', 4/7/2017 בשעה 23:59
האחראי על התרגיל:	אסף קול
	sasafca@campus

MamamScript

בתרגיל זה, נבנה סקריפט עם סינטקס דומה לזה של Matlab. הסביבה תאפשר עבודה עם מספרים שלמים בלבד ותתמוך במשתנים סקלרים ומטריצות. הקוד מבצע את פענוח הפקודות. המטלה שלכם היא להשלים את משתני הסביבה ומפת המשתנים.

שימו לב:

1. אחרי חלק 1, הכוונה במצביע היא לאובייקט מסוג MySharedPtr.
2. נתון הקובץ ScriptExceptions.h בו מוגדרים מאקרואים (define) כדי למנוע שגיאות כתיב במחרוזות הנזרקות. יש להשלים בו את המימוש הקצר של המחלקה. כאשר יתוארו החריגות בתרגיל יכתב המאקרו שיש לזרוק.
3. מומלץ לפתח את מבני התרגיל בסדר בו הם מוצגים ולבדוק אותם בנפרד ולא לממש את כולם ולבדוק אותם רק בעזרת קובץ ה main.
4. תיאור הממשק של תוכנית ה main מופיע אחרי תיאור המבנים.
5. קבצי הקוד שיש לעדכן\לשנות הם אלה המתאימים לשמות המחלקות אותם אתם מתבקשים לממש. לא מומלץ לשנות את שאר הקבצים.

MySharedPtr .1

לצורך מימוש התרגיל, נממש את המחלקה הגנרית MySharedPtr (template). מחלקה זאת מחליפה את השימוש במצביעים רגילים ומונעת דליפות זיכרון. היא מכילה מצביע מסוג האובייקט הגנרי ומצביע למונה (counter). המונה סופר את מספר האובייקטים של המחלקה אשר מצביעים לאותו זיכרון וכאשר הוא מתאפס אז הזיכרון משוחרר ע"י פעולת delete. המחלקה מעמיסה את אופרטור ה- * (dereference) אשר מחזיר reference לאובייקט המוצבע. בנוסף המחלקה מעמיסה את אופרטור ה- > ומממשת את הפעולה get אשר מחזירות את המצביע של האובייקט.

להלן קוד העושה שימוש במחלקה. ההערות מתארות את המתרחש בכל שורה:

```
int main()
{
    MySharedPtr<int> p1(new int); // p1 counter=1
    *p1 = 0; // the value pointed to by p1 is changed to 0
    cout << (*p1) << endl; // prints the value pointed to by p1 (prints 0)
    cout << (*(p1.get())) << endl; // same as previous line
    MySharedPtr<int> p2(p1); // p1/p2 counter=2
    MySharedPtr<int> p3(new int); // p3 counter =1
    p3 = p1; // p3/p2/p1 counter=3, previous p3 object deleted
    MySharedPtr<int> p4(new int); // p4 counter =1
    p2 = p4; // p2/p4 counter = 2, p1/p3 counter = 2, no object is deleted
    p1 = p2; // p1/p2/p4 counter =3, p3 counter = 1, no object is deleted
    return 0;
}
// d'tors are called at end of main
// 1. p4 d'tor => p1/p2 counter =2, no object is deleted
// 2. p3 d'tor => p3 counter =0, object deleted
// 3. p2 d'tor => p1 counter =1, no object is deleted
// 4. p1 d'tor => p1 counter =0, object deleted
```

ScriptException .2

מחלקת החרגות שתשמש אותנו לתרגיל. מאותחלת ע"י מחזירות (string) ובעלת מתודת what אשר מחזירה את המחזרות שהתקבלה באתחול. חשבו האם היא צריכה לרשת ממחלקה אחרת (ניתן להסיק זאת מקובץ ה main).

Variable/Scalar/Matrix .3

אלה הם משתני הסביבה שלנו. Variable הינה מחלקה אבסטרקטית אשר Scalar ו Matrix יורשים ממנה. היא מכילה את ממשק השימוש במשתנים כך שבהמשך נוכל להשתמש רק במצביעים מסוג Variable.

המשתנים ייתמכו בהדפסה ע"י אופרטור << של מחלקת ostream אשר מקבל מצביע ל Variable כקלט.

למשתנים יהיו את הקונסטרוקטורים הבאים:

חתימה	פעולה
Scalar(int val)	מאתחלת סקלר עם הערך val
Matrix(int row, int col, int val)	מאתחלת מטריצה עם row שורות col עמודות אשר כל ערכיה הם val
Matrix(int startVal, int endVal)	מאתחלת מטריצה עם שורה אחת (וקטור שורה) ו (endVal – startVal + 1) עמודות. הערך בעמודה הראשונה הוא startVal הערך בכל עמודה אחרי הראשונה הוא ערך הקודמת פלוס 1.

בנוסף יהיה למשתנים את הפעולות הבאות אשר לא משנות את המבנה ומחזירות מצביע מסוג Variable למבנה חדש כפי שמתואר בטבלה הבאה:

שם ופרמטרים	פלט עבור Scalar	פלט עבור Matrix
Copy()	שיכפול של המשתנה	שיכפול של המשתנה
NumElems()	Scalar עם הערך 1	Scalar שערכו כמספר אברי המטריצה
Size()	Matrix עם שורה אחת ושתי עמודות שמכילה את מימדי המשתנה (שניהם 1 במקרה של Scalar)	
Size(int dim)	Scalar עם הערך 1	אם dim גדול מ 2 או קטן מ 1 אז זורקת BAD_INPUT. מחזירה Scalar שערכו כגודל המימד המבוקש של המטריצה.
Conv(Variable pointer)	קונבולוציה בין משתנים. מימוש נתון ב Scalar.cpp	קונבולוציה בין משתנים. מימוש נתון ב Matrix.cpp
Transpose()	שכפול של המשתנה	שיכפול של המטריצה לאחר פעולת שחלוף (transpose)

המשתנים ייתמכו בשני סוגי אופרטור [] אשר מחזירים int&. אחד מקבל int והשני מקבל IdxVec (מוגדר ב Variable.h). במקרה של זריקה ייזרק INDEX_OUT_OF_RANGE. האיבר הראשון בכל משתנה הוא באינדקס 1 (ולא 0 כמו ב C ו C++).

קלט	פלט עבור Scalar	פלט עבור Matrix
int idx	זורק אם idx שונה מ 1. מחזיר את ערך הסקלר	זורק אם idx חורג ממספר אברי המטריצה או קטן מ 1. מחזיר את האיבר מספר idx במטריצה לפי column major. כלומר אם נפרס את המטריצה לווקטור עמודה אז יוחזר האיבר ה idx בווקטור זה.
IdxVec v	זורק אם אחד מערכי הווקטור שונה מ 1. מחזיר את ערך הסקלר.	זורק אם גודלו של v שונה מ 2, או v[0] גדול ממספר השורות או קטן מ 1, או v[1] גדול ממספר העמודות או קטן מ 1. מחזיר את הערך של המטריצה בשורה v[0] ועמודה v[1]

הטכניון - מכון טכנולוגי לישראל

הפקולטה להנדסת חשמל

מבוא למערכות תוכנה

המשתנים יתמכו באופרטורים הבינאריים +, *, <, >, ==, &&, ||, אשר לא משנים את המבנה ומחזירים מצביע מסוג Variable למבנה חדש כפי שמתואר בטבלה הבאה:

אופרטור	Scalar with Scalar	Scalar with Matrix	Matrix with Matrix
+	Scalar עם תוצאת החיבור	Matrix בו הסקלר נוסף לכל איבר של המטריצה המקורית	Matrix עם תוצאת חיבור המטריצות. זורק BAD_MAT_DIMS("+") אם ממדיי המטריצות לא זהים
*	Scalar עם תוצאת הכפל	Matrix בו הסקלר מכפיל כל איבר של המטריצה המקורית	Matrix עם תוצאת כפל המטריצות. זורק BAD_MAT_PROD אם ממדיי המטריצות לא מאפשרים כפל מטריות
<	Scalar עם תוצאת הפעולה הלוגית	Matrix עם תוצאת הפעולה הלוגית של הסקלר עם כל איבר של המטריצה המקורית	Matrix עם תוצאת הפעולה הלוגית בין אברי המטריצות. זורק BAD_MAT_DIMS("> / <") אם ממדיי המטריצות לא זהים
>			Matrix עם תוצאת הפעולה הלוגית בין אברי המטריצות. זורק BAD_MAT_DIMS("> / <") אם ממדיי המטריצות לא זהים
==			Matrix עם תוצאת הפעולה הלוגית בין אברי המטריצות. זורק BAD_MAT_DIMS("==") אם ממדיי המטריצות לא זהים
&& 			זורק BAD_INPUT

VariablesMap .4

בחלק זה מומלץ להשתמש במחלקות map ו vector (פחות חשוב) של Map<T,S>.stl הינו מיפוי חד ערכי בין ערכים מ T לערכים ב S. כלומר, לכל ערך ב T יש (לכל היותר) ערך יחד ב S אליו הוא ממופה אך ההפך איננו בהכרח נכון. כלומר, יכולים להיות מספר ערכים ב T אשר ממפים לאותו ערך ב S. VariablesMap הינו מיפוי בין מחזריות (string) למצביעים מסוג Variable. במידה ואינכם משתמשים במחלקת map של stl לצורך המימוש אז מומלץ לממש בעצמכם מחלקה גנרית בשם MyMap אשר תהווה גרסה מרודדת של מחלקת map ולהשתמש בה לצורך מימוש VariablesMap.

המחלקה VariablesMap תתמוך בפעולות הבאות:

חתימה	פעולה
VariablesMap()	מאתחל מפת משתנים ריקה (עם שם שמור שאיננו שם משתנה חוקי)
SetSavedName(const string& newSavedName)	מעדכנת את שם המשתנה השמור
GetTmpVariable()	מוצאת שם משתנה פנוי (אשר עדיין לא קיים במיפוי ובנוסף איננו השם השמור). יוצרת משתנה זמני חדש בשם זה ומוסיפה אותו למיפוי. מחזירה את שם המשתנה הזמני החדש
ClearTmpVars()	מוחקת את כל המשתנים הזמניים מהמיפוי
Operator[] (const string& x)	בודקת האם x הוא שם משתנה חוקי (מתחיל באות אנגלית וכל שאר התווים שייכים ל VALID_VAR_CHARS (נתון ב VariablesMap.h). אם לא אז זורקת INVALID_VAR_NAME(x). אחרת מייצרת מיפוי חדש עם מפתח שערכו x ומחזירה reference אליו.
at(const string& x)	מחזירה reference לערך אליו x ממפה. במידה ולא קיים מיפוי כזה אז זורקת std::out_of_range (חריגה מ stdexcept)
erase(const string& x)	מסירה את המיפוי שמתאים ל x מהמבנה

הטכניון - מכון טכנולוגי לישראל
הפקולטה להנדסת חשמל
מבוא למערכות תוכנה

5. תוכנית ה main

שאר הקוד בתרגיל נתון לכם וכל שנשאר הוא להריץ (ואולי להנות מהתוצאה?). התוכנית תומכת במספר פקודות דמויות matlab. באופן כללי היא אמורה להיות חסינה לשגיאות כך ששימוש ב syntax לא תקין רק יגרום להדפסת הודעת שגיאה מתאימה למסך. שורות הפקודה האפשריות הן:

פקודה	הסבר
if(<expression>)	ביצוע מותנה של השורות הבאות עד לסיום הבלוק (ע"י שורת end)
for <var name> = <expression>	לולאה for אשר בה <var name> הוא משתנה הלולאה ותוצאת <expression> היא הערכים אותם יקבל המשתנה. לדוגמא 1:5 כ <expression> יגרום לכך שבאיטרציה הראשונה משתנה הלולאה יהיה 1 בשניה 2 וכך עד 5. בדומה ל if בלוק הלולאה מסתיים בשורת end.
<var name> = <expression>	מציב לתוך <var name> את תוצאת הביטוי <expression>. לדוגמא a=1:5 מייצר משתנה ווקטור שורה (מטריצה) בשם a ושערכיו 1 עד 5. ניתן גם להציב לתוך תת אינקסים של משתנה קיים. לדוגמא a(2:3)=7:8 ישנה את הערכים השני והשלישי ב a ל 7 ו 8 בהתאמה
disp <var name>	מדפיס למסך את המשתנה ששמו <var name> (במידה וקיים כזה)

כאשר ניתן לקנן לולאה וביטויי if (אך לא לשכוח לסגור את הבלוקים של כל אחד בסוף). הביטויים מהסוג <expression> בטבלה יכולים להכיל ערבוב (תקין) של הפעולות הבאות:

פעולה	הסבר
numel(<var>)	מחזיר את מספר האלמנטים של <var>
size(<var>)	מחזיר ווקטור שורה עם מימדי <var>
size(<var>,<dim>)	מחזיר את המימד ה <dim> של <var>
conv(<var1>,<var2>)	מחזיר את תוצאת הקובולוציה בין <var1> ל <var2>
zeros(<dims>)	מחזיר מטריצת אפסים שמימדיה נתונים ב <dim>. אם נתון רק מימד אחד אז נוצרת מטריצה עם <dim> שורות ועמודות
transpose(<var>)	מחזיר את תוצאת השחלוף (transpose) של <var>
<var> (<indices>)	מחזיר את תת המטריצה של <var> לפי האינדקסים הלינארים הנתונים ב <indices>
<var>(<row idxs>,<col idxs>)	מחזיר את תת המטריצה של <var> לפי אינדקסי העמודה והשורה הנתונים ב <row idxs> ו <col idxs> בהתאמה
(<expression>)	סוגריים לצורך שינוי סדר קדימויות בביצוע הפעולות
<var1>*<var2>	מחזיר את תוצאת הכפל בין <var1> ל <var2>
<var1>+<var2>	מחזיר את תוצאת החיבור בין <var1> ל <var2>
<var1>:<var2>	מחזיר ווקטור שורה עם ערך באינדקס הראשון ששווה ל <var1> ואשר כל ערך אחריו גדל ב 1 ביחס לקודמו עד הערך <var2>
<var1> < <var2>	מחזיר את תוצאת < בין <var1> ל <var2> (0 לשקר ו 1 לאמת)
<var1> == <var2>	מחזיר את תוצאת == בין <var1> ל <var2> (0 לשקר ו 1 לאמת)
<var1> && <var2>	מחזיר את תוצאת AND בין <var1> ל <var2> (0 לשקר ו 1 לאמת)
<var1> <var2>	מחזיר את תוצאת OR בין <var1> ל <var2> (0 לשקר ו 1 לאמת)

הביטויים המצורה <????> בטבלה זאת יכולים להיות ערבוב של הביטויים האחרים, של שמות משתנים, ושל מספרים (חיוביים ושליליים).

במידה והסקריפט נתקל בשגיאה אז הוא מדפיס הודעת שגיאה ועוצר כל פעולה מתמשכת כמו לולאה או if ומתחיל את הפענוח מחדש בשורה הבאה.

6. תרגילון Bash

כתבו סקריפט בשם `change_extensions`, בן 8 שורות לכל היותר (לא כולל `#!/bin/bash` השורה) ולא כולל שורות תיעוד) המקבל שלושה פרמטרים. שם תיקייה, ושני סיומות קבצים. הסקריפט משנה את הסיומות של כל הקבצים בתיקייה שהתקבלה הנגמרים בסיומת שהתקבלה כפרמטר השני, לסיומת שהתקבלה כפרמטר השלישי.
לדוגמה, השורה

```
change_extensions dir cpp C
```

תשנה את הסיומות של כל הקבצים בתיקייה `dir` שנגמרים ב-`cpp` לסיומת `C`. אין צורך לבצע בדיקת פרמטרים.

הערה: קראו באינטרנט על מציאת תבניות במשתנים באמצעות האופרטורים

```
${var#Pattern}  
${var##Pattern}  
${var%Pattern}  
${var%%Pattern}
```

למשל, ניתן לקרוא על כך כאן:

<http://tldp.org/LDP/abs/html/parameter-substitution.html>

הערה 2: הסקריפט הנ"ל שימושי כאשר מעבירים קבצים שנכתבו ב-windows ל-unix.

7. הנחיות הגשה

1. קבצי קוד חלקיים, וכן קבצי קלט ופלט לדוגמה, נמצאים בתיקייה:

```
~eesoft/hmw/hmw5
```

לפני תחילת העבודה, הורידו את הקבצים לחשבונכם באמצעות הפקודה:

```
cp ~eesoft/hmw/hmw5/* .
```

2. עברו היטב על הוראות ההגשה של תרגילי הבית המופיעים באתר טרם ההגשה! ודאו כי התכנית שלכם עומדת בדרישות הבאות:

- התכנית קריאה וברורה
 - התכנית מתועדת היטב לפי דרישות התיעוד המופיעות באתר
 - התכנית מתקמפלת עם הדגלים `-Wall -std=c++0x -g` ללא שגיאות וללא warnings כלל
 - התכנית רצה ללא דליפות זיכרון וגישות לא חוקיות לזיכרון כלל (בדיקה באמצעות `valgrind`)
 - התכנית נותנת פלט זהה לחלוטין לפלט הצפוי על כל קבצי הקלט שסופקו (בדיקה באמצעות פקודת `diff` על קבצי הפלט)
 - קובץ ה-`makefile` יוצר קובץ הרצה בשם הנדרש
3. יש להגיש קובץ `tar` יחיד המכיל את כל הקבצים שאתם נדרשים להגיש ואותם בלבד – ללא תתי-תיקיות. ודאו כי לא שכחתם את קובץ `readme` המכיל את פרטי הסטודנטים, וכן את ה-`makefile` במידה ונדרשתם.
4. שאלות בנוגע לתרגיל יש להפנות לפורום התרגיל ב-moodle בלבד – ניתן לשלוח שאלות במייל למתרגל האחראי על התרגיל בלבד, ורק במידה והשאלה מכילה פתרון חלקי.
5. סיכום מפרט התרגיל:

הטכניון - מכון טכנולוגי לישראל
 הפקולטה להנדסת חשמל
 מבוא למערכות תוכנה

סעיף		תיאור
נושא התרגיל		C++ מסכם
תאריך ההגשה		יום ג', 4/7/2017 בשעה 23:59
המתרגל האחראי על התרגיל		אסף קסל
תיקייה המכילה קבצים לשימוש הסטודנטים		sasafca@campus ~eesoft/hmw/hmw5
קבצי הקוד הנתונים		main.cpp Variable.h Commands.h Variable.cpp Commands.cpp Scalar.h LineParser.h Scalar.cpp LineParser.cpp Matrix.h MamatScriptEngine.h Matrix.cpp MamatScriptEngine.cpp VariablesMap.h ScriptExceptions.h VariablesMap.cpp MySharedPtr.h
קבצי הקלט והפלט הנתונים		1.MamatScript.in 1. MamatScript.out
הקבצים שיש להגיש		readme ScriptExceptions.h makefile MySharedPtr.h change_extensions Variable.h main.cpp Variable.cpp Commands.h Scalar.h Commands.cpp Scalar.cpp LineParser.h Matrix.h LineParser.cpp Matrix.cpp MamatScriptEngine.h VariablesMap.h MamatScriptEngine.cpp VariablesMap.cpp
שם תכנית ההרצה הדרושה (הנוצרת ע"י makefile)		MamatScript
דגשים מיוחדים		

בהצלחה!