

תרגיל בית 5 – c++

מועד ההגשה:	2/7/2020
האחראי על התרגיל:	תום נורמן
tom.norman@campus.technion.ac.il	

WolframMamat

שימו לב: קראו את הוראות ההגשה היטב. בפרט, הקפידו על הדגלים לפקודת הקמפול.

בתרגיל זה נממש מחשבון פונקציות בדומה למנוע המתמטי wolfram alpha. המנוע שלנו ייקרא WolframMamat ויתמוך בפולינומים והרכבת פונקציות.

הפעולות בהן נתמוך הן 3 פעולות חשבון: חיבור, חיסור וכפל.

תזכורות:

1. פולינום $a(x)$ מסדר n הוא פונקציה מהצורה $a(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$,

כאשר a_i הם המקדמים. אנחנו נניח לכל אורך התרגיל שכל המקדמים הם שלמים (כלומר int).

a. נגזרת: $a'(x) = a_1 + 2a_2x + \dots + na_nx^{n-1}$

b. אינטגרל: $\int a(x)dx = C + a_0x + \frac{a_1}{2}x^2 + \frac{a_2}{3}x^3 + \dots + \frac{a_n}{n+1}x^{n+1}$

2. הרכבה של פונקציה $f(x)$ על פונקציה $g(x)$ היא מהצורה $f(g(x))$.

הערות:

1. לאורך כל התרגיל, המנוע שלנו יעבוד אך ורק עם מספרים שלמים (int).
2. מותר להשתמש בכל יכולות STL.
3. בנוסף ניתן לעשות include רק לספריות שמתחילות בstd וב-math.

במערכת התוכנה שלנו יהיו 3 מחלקות: func, polynom, compfunc:

חלק א: מחלקת func

מחלקת func מהווה בסיס למחלקות האחרות והיא תהיה מחלקה אבסטרקטית מכיוון שאופן פעולתה (איך מחשבים פלט $f(x)$ בהינתן קלט x) לא מוגדר. בחלק זה תממשו את המחלקה הנ"ל בקבצים func.h ו-func.cpp. הצהרה חלקית שלה ומימוש חלקי שלה נתונים לכם.

על המחלקה לתמוך בכל הפעולות אשר נדרשות ממנה בקובץ main.cpp (ראה בהמשך):

1. `func& operator<<(const int& x)`
 - a. בקצרה: הכנסת ערך חדש
 - b. קלט: ערך שיש להכניס לפונקציה (x)
 - c. פלט: הפונקציה עצמה (אפשרות לשרשור אופרטורים)
 - d. פעולה: שומרת את הזוג $(x, f(x))$ ב-`fmap_`.
 - e. דוגמא לקריאה: ראה בקובץ main.

2. `ostream& operator<<(ostream&, const func&)`
 - a. בקצרה: אופרטור הדפסה
 - b. קלט:
 - i. שמאל: ערוץ פלט
 - ii. מימין: הפונקציה אותה רוצים להדפיס
 - c. פלט: ערוץ פלט (אפשרות לשרשור אופרטורים)
 - d. פעולה: מדפיסה את הפונקציה (פורמטים בהמשך)
 - e. דוגמא לקריאה: ראה בקובץ main.

3. עוד מתודות ומשתנים פרטיים לפי ראות עיניכם על מנת שהמחלקה תעבוד כמו שצריך.

קוד שמסופק לכם:

1. func.h – הצהרה חלקית:

```
class func {
protected:
    int maxVal_;
    int minVal_;
    map<int,int> fmap_;
    void plot(ostream& os) const ;
};
```

2. func.cpp – מימוש של מתודת הדפסת גרף הפונקציה

```
void plot(ostream& os) const ;
```

maxVal, minVal נדרשים ליצירת plot של הפונקציה והם מכילים את הערכים המקסימליים והמינימליים שמתקבלים ע"י `func& operator<<(const int& x)` (באחריותכם לאתחל ולעדכן אותם).

<http://www.cplusplus.com/reference/map/map>

חלק ב: מחלקת polynom

מחלקת polynom יורשת ממחלקת func והיא מממשת פונקציה שהינה פולינום. בחלק זה תממשו את המחלקה הנ"ל בקבצים polynom.h ו-polynom.cpp. הצהרה חלקית שלה ומימוש חלקי של המחלקה נתונים לכם. על המחלקה לתמוך בכל בפעולות הבאות:

1. בנאי

- a. קלט: סדר הפולינום ומצביע למערך המקדמים
- b. פעולה: מאתחל אובייקט מסוג פולינום
- c. דוגמא לקריאה:

```
int arr[4] = {1,1,0,-1};  
polynom p(3,arr); //  $p(x) = 1 + x - x^3$ 
```

אין להניח שאחרי איתחול פולינום המצביע למערך המקדמים ימשיך להצביע לאותו מערך.

2. חיבור, חיסור וכפל

- a. בקצרה: ביצוע פעולות חשבון על פולינומים
- b. פלט: פולינום חדש שהוא תוצאה החיבור/חיסור/כפל
- c. דוגמא לקריאה: ראה קובץ main

3. נגזרת

- a. בקצרה: ביצוע נגזרת לפי הנוסחא למעלה
- b. פלט: פולינום הנגזרת.
- c. דוגמא לקריאה:

```
polynom q=p.Derivative(); //  $q(x) = 1 - 3x^2$ 
```

4. אינטגרל

- a. בקצרה: ביצוע אינטגרל לפי הנוסחא למעלה
- b. פלט: פולינום האינטגרל
- c. דוגמא לקריאה:

```
polynom a=q.Integral(); //  $a(x) = x - x^3$ 
```

5. הדפסה

- a. פורמט לדוגמא:

```
q<<-1<<0<<1; // 3 points saved: (-1,-2),(0,1),(1,-2)
```

```
cout<< "q(x)="<< q <<endl;
```

הטכניון - מכון טכנולוגי לישראל
הפקולטה להנדסת חשמל
מבוא למערכות תוכנה

$q(x) = -3x^2 + 1$
Derivative: $-6x$
Integral: $-x^3 + x + C$
1 *
0
-1
-2 * *
-1 0 1

b. הערות:

- i. השורות 4 עד הסוף מודפסות על ידי קוד שמסופק לכם.
- ii. ראו את קבצי הפלט לעוד דוגמאות.

6. עוד מתודות ומשתנים פרטיים לפי ראות עיניכם על מנת שהמחלקה תעבוד כמו שצריך.

קוד שמסופק לכם:

1. polynom.h - הצהרה חלקית של המחלקה

```
class polynom : public func {
public:
    void printcoefs(ostream&) const ;
protected:
    int n_; //order of the polynom
    int* coefs_; //coefficients
};
```

2. polynom.cpp – מימוש של פונקציית הדפסת מקדמי הפולינום.

הערות: ניתן להשתמש בספריית math על מנת לחשב חזקה בעזרת פונקציית הספרייה pow.

חלק ג: מחלקת compfunc

מחלקת compfunc יורשת ממחלקת func והיא מממשת פונקציית הרכבה. בחלק זה תממשו את המחלקה הנ"ל בקבצים compfunc.h ו-compfunc.cpp.

על המחלקה לתמוך בכל בפעולות הבאות

1. בנאי

a. קלט: פונקציה חיצונית ופונקציה פנימית

b. פעולה: מאתחל אובייקט מסוג פונקציית הרכבה

c. דוגמא לקריאה:

```
compfunc f(q,p); //  $f(x) = q(p(x))$ 
```

2. הדפסה

a. פורמט:

```
f<< -1<<0<<1; //3 points saved: (-1,-2),(0,-2),(1,-2)
```

```
cout<<"q(p)(x)=" <<f <<endl;
```

```
q(p)(x)=  
-2* * *  
-1 0 1
```

b. הערות:

i. אין צורך בנגזרת או אינטגרל

3. עוד מתודות ומשתנים פרטיים לפי ראות עיניכם על מנת שהמחלקה תעבוד כמו שצריך.

חלק ד: קובץ החינה main ומבנה קובץ הקלט

בתרגיל זה, נתון לכם קובץ main.cpp. **אין לשנות קובץ זה!** החינה מבצע את הפעולות הבאות:

- קריאת פרמטרים (ראה למטה) ופקודות מהקלט הסטנדרטי
- ביצוע הפעולות מקובץ הקלט

קבצי הקלט מכילים שורות מהצורה הבאה:

FUNC_CMD FUNC_NAME FUNC_VALUE

כאשר FUNC_CMD זאת הפקודה שיש להריץ, FUNC_NAME זה שם הפקונציה המתאימה לפעולה, ו-FUNC_VALUE זה פרמטר שמועבר לצורך הפעולה.

הפקודות הנתמכות:

הערות	FUNC_VALUE	בקצרה	FUNC_CMD
מובטח שסדר הפוליונים מתאים לכמות המקדמים. סדר הפוליונים מגיע ראשון ואחריו המקדמים מופרדים בפסיק	1. סדר הפוליונים – מספר טבעי 2. מקדמים – מספרים שלמים	הכנסת פוליונים לזיכרון המנוע	SetPolynom
נקודות החישוב מופרדות בפסיק	הכנסת נקודות חישוב לפונקציה – מספרים שלמים	הכנסת נקודה לפונקציה	AddPoint
ייתכן ופוליונים א' או ב' לא הוכנסו עדיין. קובץ החינה main יטפל בזה וידיפס הודעה. אם הם כן נמצאים, אז מובטח שהם פוליונים.	1. שם פוליונים א' 2. פעולת חישוב: +, -, * 3. שם פוליונים ב'	ביצוע פעולות חישוב על פוליונים	MathPolynom
יתכן כי אחת הפונקציות לא הוכנסה עדיין. קובץ החינה main יטפל בבעיה זו.	1. שם פונקציה חיצונית 2. שם פונקציה פנימית	הכנסת פונקציית הרכבה לזיכרון המנוע	SetCompfunc
קובץ החינה main דואג לחפש פונקציה, או להדיפס לפי סדר.	FUNC_NAME: או שזה שם של פונקציה אותה מדפיסים או שמועבר הפרמטר All שאומר להדיפס את כל הפונקציות לפי סדר לקסיקוגרפי של שמותיהן.	הדפסה	Print

הטכניון - מכון טכנולוגי לישראל
הפקולטה להנדסת חשמל
מבוא למערכות תוכנה

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 התייעוד המופיעות באתר
 - התכנית מתקמפלט ללא שגיאות וללא warnings כלל

i. **שימו לב:** הדגלים של ++g הם g++-std=c++11, -Wall, -g. בלבד. לפתרונות אשר לא יתקמפלו עם הדגלים האלו בלבד יורדו נקודות!

 - 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.

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:

סעיף				תיאור
נושא התרגיל				C++ חפיפת אופרטורים, חריגות ופולימורפיזם.
תאריך ההגשה				2/7/2020
המתרגל האחראי על התרגיל				תום נורמן tom.norman@campus.technion.ac.il
תיקייה המכילה קבצים לשימוש הסטודנטים				~eesoft/hmw/hmw5
קבצי הקוד הנתונים				main.cpp
				polynom.h
				func.h
				func.cpp
קבצי הקלט והפלט הנתונים				output1
				input1
				output2
				input2
הקבצים שיש להגיש				readme
				makefile
				compfunc.h
				compfunc.cpp
				polynom.h
				polynom.cpp
				func.h
				func.cpp
שם תכנית ההרצה הדרושה (הנוצרת ע"י makefile)				WolfraMamat
דגשים מיוחדים				שימו לב: הדגלים של ++g הם g++-std=c++11, -Wall, -g. בלבד!

בהצלחה!