

מעבדה 3 : BigInt ADT (Abstract Data Type)

מעבדה זו יש להגיש גם מודפסת וגם במודל עד יום ה' 21.11 בשעה 10:00.
את העבודה המודפסת יש להגיש בתא של יעל 548EM

המשימות:

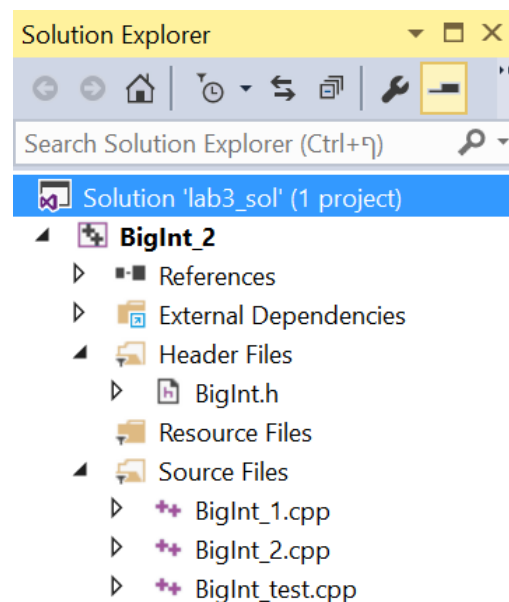
- פרוייקט עם מספר קבצים
- זיהוי וטיפול בדליפות זיכרון (memory leaks)
- operator overload
- מימוש הפונקציות פיבונצ'י ושורש על BigInt

פרוייקט עם מספר קבצים:

פרוייקט יכול להכיל פונקציות רבות ושורות רבות של קוד. מקובל לחלק את הקוד לקבצים:
קבצי header (h) – הגדרות של טיפוסים, חותמות של פונקציות.
קבצי מימוש (cpp) – מימוש הפונקציות, תוכניות בדיקה.
קבצי המימוש מכילים את קבצי ההגדרות ע"י "include"

חלק את הפרוייקט BigInt ל 4 קבצים, קמפלו והריצו:
BigInt.h – הגדרת המבנה וחותמות הפונקציות
BigInt_1.cpp – מימוש הפונקציות create, print, compare, assign
BigInt_2.cpp – מימוש הפונקציות add, inc
BigInt_test.cpp – מימוש main

ה Solution Explorer אמור להראות כך:



קבצי header

כדי למנוע מצב של הגדרה כפולה במקרה שעושים include פעמיים לאותו הקובץ,
בראש כל header כותבים *include guards*:

```
#ifndef _UNIQUE_NAME_H_
#define _UNIQUE_NAME_H_

#endif
```

ובסוף כל header כותבים:

זיהוי וטיפול בדליפות זיכרון (memory leaks):

בראש כל קובץ cpp (או בראש קובץ header בו משתמשים) יש להוסיף את השורות:

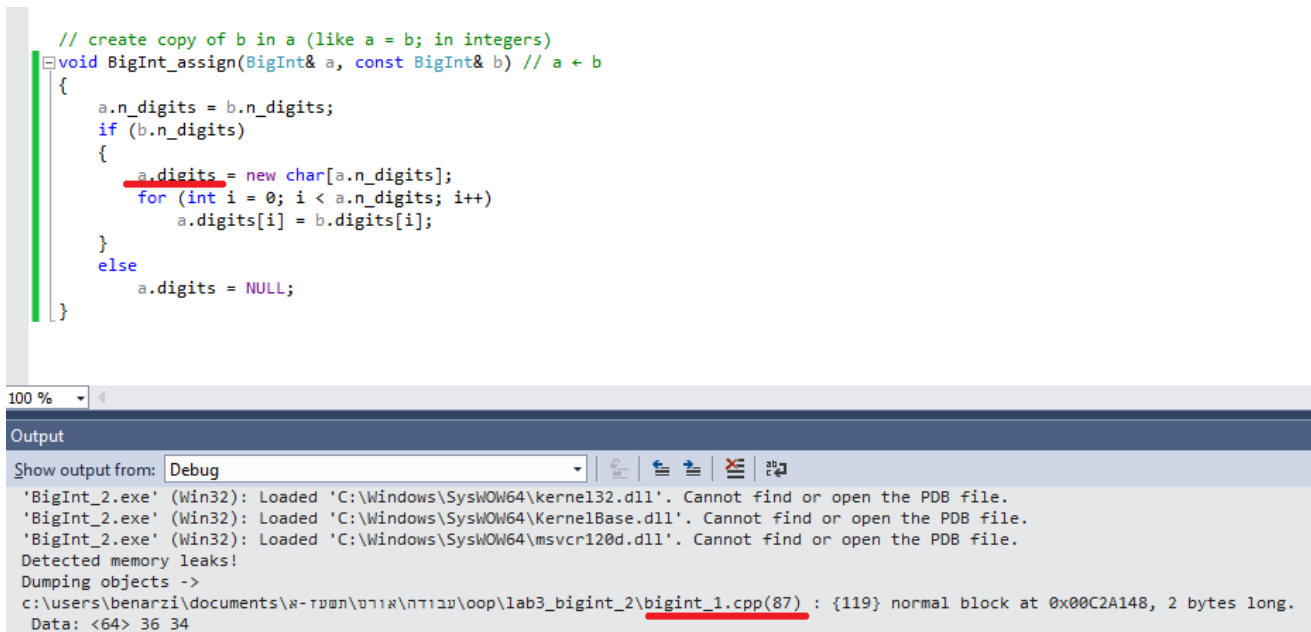
```
#define _CRTDBG_MAP_ALLOC
#include <stdlib.h>
#include <crtdbg.h>
#define DBG_NEW new ( _NORMAL_BLOCK , __FILE__ , __LINE__ )
#define new DBG_NEW
```

כדי לבדוק דליפת זכרון בכל שלב בתוכנית יש לקרוא ל:

```
_CrtDumpMemoryLeaks();
```

ההודעות המפורטות על דליפות הזכרון מופיעות בחלון ה output.

דוגמא: בקובץ BigInt_1.cpp יש דליפת זיכרון של 2 בתים בשורה 87.



```
// create copy of b in a (like a = b; in integers)
void BigInt_assign(BigInt& a, const BigInt& b) // a ← b
{
    a.n_digits = b.n_digits;
    if (b.n_digits)
    {
        a.digits = new char[a.n_digits];
        for (int i = 0; i < a.n_digits; i++)
            a.digits[i] = b.digits[i];
    }
    else
        a.digits = NULL;
}
```

Output

Show output from: Debug

'BigInt_2.exe' (Win32): Loaded 'C:\Windows\System32\kernel32.dll'. Cannot find or open the PDB file.
'BigInt_2.exe' (Win32): Loaded 'C:\Windows\System32\kernelbase.dll'. Cannot find or open the PDB file.
'BigInt_2.exe' (Win32): Loaded 'C:\Windows\System32\msvcrt120d.dll'. Cannot find or open the PDB file.
Detected memory leaks!
Dumping objects ->
c:\users\benarzi\documents\א-טבוכדה\אורט\מסעז\lab3_bigint_2\bigint_1.cpp(87) : {119} normal block at 0x00C2A148, 2 bytes long.
Data: <64> 36 34

תקנו את מעבדה 2 לעבוד עם reference כך שאלו החתימות:

```
void BigInt_print(const BigInt& bi);    //print number to the screen
```

```
BigInt BigInt_create(int num);    //creat from int
```

```
int BigInt_compare(const BigInt& a, const BigInt& b);    //-1 if a<b  
// 0 if a = b  
// 1 if a>b
```

```
// create copy of b in a (like a = b; in integers)
```

```
void BigInt_assign(BigInt& a, const BigInt& b);    // a ← b
```

```
// Free memory dynamically allocated
```

```
void BigInt_destroy(BigInt& bi);
```

```
int main()
{
    //create from C string
    BigInt a = BigInt_create("0009999999999999999999");
    BigInt_print(a);
    cout << endl;

    //creat from int
    BigInt b = BigInt_create(9345);
    BigInt_print(b);
    cout << endl;

    // compare
    int i = BigInt_compare(a, b);
    BigInt_print(a);
    switch (i)
    {
        case 0: cout << " == ";
                break;
        case 1: cout << " > ";
                break;
        case -1: cout << " < ";
                 break;
        default: cout << "???" ;
    }
    BigInt_print(b);
    cout << endl;

    // assign
    BigInt_assign(b, a);
    BigInt_print(b);
    cout << endl << "After assigning b to a, compare = ";
    cout << BigInt_compare(a, b) << endl;

    // inc
    BigInt_print(a);
    cout << " + 1 = ";
    BigInt_inc(a);
    BigInt_print(a);
    cout << endl;

    // add
    BigInt c = BigInt_create(93652);
    BigInt d = BigInt_add(a, c);
    BigInt_print(a);
    cout << " + ";
    BigInt_print(c);
    cout << " = ";
    BigInt_print(d);
    cout << endl;

    BigInt_destroy(a);
    BigInt_destroy(b);
    BigInt_destroy(c);
    BigInt_destroy(d);

    cout << "Leaks: " << _CrtDumpMemoryLeaks() << endl;

    return 0;
}
```

מימוש הפונקציות פיבונצ'י ושורש על BigInt:

עם מערכת הפונקציות שכתבת עבור מבנה הנתונים BigInt כתוב ובדוק (כולל העדר דליפות זכרון) את הפונקציות הבאות.

ממש את הפונקציות בקובץ נפרד BigInt_app.cpp.

הוסיפו את בדיקת הפונקציות החדשות ל-main שבקובץ BigInt_test.cpp

```
//The n'th Fibonacci number n(1)=1,n(2)=1,n(3)=2,n(4)=3,n(5)=5,...
BigInt fibo(unsigned int n);

//sqrt(x)
BigInt sqrt(const BigInt& x);
```

את הפונקציה לחישוב השורש כתוב בשטת "החיסורים" לפי האלגוריתם הבא:

Square_root(x)

```
1 sum=0, odd=1, sq=0
2 while sum < x
3     sum=sum+odd
4     sq++
5     odd=odd+2
6 return sq
```

Operator Overload

כעת נחליף שתי חתימות של פונקציות בחתימה עם האופרטור המתאים:

1. החליפו את הפונקציה BigInt_compare בפונקציה שחתימתה:

```
int operator==(const BigInt& a, const BigInt& b);
```

2. החליפו את הפונקציה BigInt_add בפונקציה שחתימתה:

```
BigInt operator+(const BigInt& a, const BigInt& b);
```

תקנו את הקוד שלכם שיעבוד עם החתימות החדשות.

😊 בהצלחה

אופן ההגשה

1. הגשה בבודדים
2. העבודה תוגש מודפסת ומשודכת עם הקבצים:
BigInt_test.cpp, BigInt_app.cpp, BigInt_2.cpp, BigInt_1.cpp, BigInt.h
3. על התוכנית להיות כתובה באופן מסודר ומדורג (הזזה ימינה עם פתיחה של כל סוגריים מסולסים).
4. יש להוסיף הערות הסבר באנגלית בתוכנית.
5. בנוסף להדפסות יש להגיש את כל קבצי הקוד שכתבתם במודל:
a. על הקבצים להיות מכווצים בתוכנת zip ולא בשום תוכנת כיווץ אחרת.
b. שם הקובץ zip יהיה ת.ז. של הסטודנטים כך: id1_id2.zip.