



אוניברסיטת אריאל בשומרון

פקולטה: מדעי הטבע

מחלקה: מדעי המחשב

שם הקורס: מבנה זיכרון ושפת C++

קוד הקורס: 2-7027810 כל הקבוצות - קבוצות 1,2,3,4

מבחן לדוגמה

משך הבחינה: 3 שעות

שם המרצים: אראל סגל-הלוי, חרות סטרמן

יש לענות על כל השאלות במחברת הבחינה ולהגיש את המחברת בלבד - אין להגיש את הטופס.

אין להעתיק את השאלות מהטופס למחברת הבחינה. טופס הבחינה יתפרסם במודל מיידי לאחר הבחינה.

בשאלות 1-5, כששואלים "מה סוג השגיאה", יש לכתוב אחת מהאפשרויות הבאות:

- שגיאת קומפילציה (compilation error)
- שגיאת קישור (link error)
- שגיאת זמן-ריצה (runtime error)
- שגיאה לוגית (logic error)
- דליפת זיכרון (memory leak).

כששואלים "מה גורם לשגיאה" יש לציין מספרי שורות בהתאם לקוד המובא בשאלה.
כששואלים "איך לתקן את השגיאה" יש לציין מספרי שורות ולהסביר מה צריך לשנות בכל שורה/שורות.
בחלק מהמקרים ישנם כמה תיקונים אפשריים. ניקוד מלא יינתן רק לתיקון הטוב ביותר מנקודת מבט של הנדסת תוכנה, כפי שנלמד בכיתה.

יש לענות תשובות מלאות וממוקדות. לא יינתנו נקודות על תשובות עם טקסט מיותר שאינו קשור לנושא.

ניתן לענות על כל שאלה במילים "לא יודע" או "לא יודעת". במקרה זה השאלה לא תיבדק כלל, ואתם תקבלו 20% מהניקוד על אותה שאלה (מעוגל כלפי מטה).

אסור להשתמש בכל חומר עזר.

בהצלחה!!!

שאלה 1 [10 נקודות]

נתונה התוכנית:

```
01  #include <iostream>
02  #include <fstream>
03  using namespace std;
04
05  class Picture {
06      char* pixels; // grayscale image - each pixel is one byte.
07      int width, length;
08  public:
09      Picture(int width, int length):
10          width(width), length(length),
11          pixels(new char[width*length]) {}
12
13      void setBackground() {
14          for (int i=0; i<width; i++) {
15              for (int j=0; j<length; j++) {
16                  pixels[i*width+j] = 65;
17              }
18          }
19      }
20
21      void write(ostream& file) {
22          file.write(pixels, width*length);
23      }
24  };
25
26  int main() {
27      Picture pic1(500,100);
28      pic1.setBackground();
29      ofstream outfile("test.bin");
30      pic1.write(outfile);
31      Picture pic2(100,500);
32      pic2.setBackground();
33  }
```

כששולחים את התוכנית לבדקן האוטומטי, מקבלים את ההודעה:
timeout: the monitored command dumped core.

- א [1 נק']. מה סוג השגיאה? (קומפילציה / קישור / זמן-ריצה / לוגית / דליפת-זיכרון).
ב [4 נק']. מה גורם לשגיאה?
ג [5 נק']. הציעו שינוי שאפשר לבצע במחלקה Picture, כך ששגיאות מסוג זה יגרמו לזריקת חריגה עם הודעת-שגיאה משמעותית יותר, כך שיהיה קל יותר לזהות אותן.

שאלה 2 [10 נקודות]

נתונה התוכנית:

```
01  #include <vector>
02  #include <iostream>
03  #include <sstream>
04  #include <cassert>
05  using namespace std;
06
07  class Complex {
08      double _re, _im;
09  public:
10      Complex(double _re, double _im): _re(_re), _im(_im) { }
11
12      double& re() { return _re; }
13      double& im() { return _im; }
14
15      double re() const { return _re; }
16      double im() const { return _im; }
17  };
18
19  istream& operator>> (istream& in, const Complex& c) {
20      in >> c.re();
21      char plus; in >> plus; if (plus!='+')
22          throw "Wrong format - missing plus sign after real part!";
23      in >> c.im();
24      char i; in >> i; if (i!='i')
25          throw "Wrong format - missing i after imaginary part!";
26      return in;
27  }
28
29  void test() {
30      istringstream input("5+7i");
31      Complex c(0,0);
32      input >> c;
33      assert(c.re()==5);
34      assert(c.im()==7);
35  }
36
37  int main() {
38      test();
39  }
```

כשמריצים make מתקבלת ההודעה הבאה:

```
main.cpp:20:5: error: invalid operands to binary expression  
( 'std::istream' (aka 'basic_istream<char>') and 'double')  
in >> c.re();  
~~ ^ ~~~~~
```

א [1 נק']. מה סוג השגיאה? (קומפילציה / קישור / זמן-ריצה / לוגית / דליפת-זיכרון).

ב [4 נק']. מה גורם לשגיאה?

ג [5 נק']. יש לתקן את התוכנית ע"י שינוי/הוספה של שורה אחת. איזה שינוי יש לבצע כך שהתוכנית תרוץ בלי שגיאות?

שאלה 3 [10 נקודות]

נתונה התוכנית:

```
02  #include <iostream>  
04  using namespace std;  
05  
06  struct Date {  
07      int _year;  
08      int _month;  
09      int _day;  
10  };  
11  
12  int main() {  
13      Date* dates = new Date[10];  
14      cout << dates[0]._year;  
15  }
```

הניחו שגודל של מספר שלם הוא 4 בתים ושל פוינטר 8 בתים.

אם לדעתכם אף תשובה אינה נכונה, כיתבו את התשובה הקרובה ביותר לתשובה הנכונה.

א. התוכנית כתובה בשפת C++. כמה בתים היא צורכת על הערימה? נמקו.

ב. בפלט של `valgrind --tool=memcheck` מופיעה השורה הבאה, כאשר N הוא מספר שלם:
`==10546== definitely lost: N bytes in 1 blocks`

מהו N? נמקו.

ג. המחלקה Date תורגמה לשפת Java, והתוכנית הראשית שונתה בהתאם ל:

```

Date[] dates = new Date[10];
for (int i=0; i<10; ++i)
    dates[i] = new Date();
System.out.println(dates[0].year());

```

כמה בתים היא צורכת על הערימה? נמקו.

ד. התוכנית שונתה ועכשיו היא נראית כך (השינויים מודגשים):

```

02  #include <iostream>
04  using namespace std;
05
06  class Date {
07      int _year;
08      int _month;
09      int _day;
10  public:
11      int& year() { return _year; }
12      int year() const { return _year; }
13  };
14
15  int main() {
16      Date* dates = new Date[10];
17      cout << dates[0].year();
18  }

```

התוכנית כתובה בשפת C++. כמה בתים היא צורכת על הערימה? נמקו.

ה. התוכנית תורגמה לשפת Java כמו בשאלה 13. כמה בתים היא צורכת על הערימה? נמקו.

שאלה 4 [10 נקודות]

נתונה התוכנית:

```
01  #include <iostream>
02  using namespace std;
03
04  class Buffer {
05      double* doubles;
06      int size;
07  public:
08
09      Buffer(int size): size(size), doubles(new double[size]) {
10          for (int i=0; i<size; ++i) doubles[i] = 0.0;
11      }
12
13      ~Buffer() {
14          delete[] doubles;
15      }
16
17      void fill(double d) {
18          Buffer newBuffer(size);
19          for (int i=0; i<size; ++i) newBuffer.doubles[i] = d;
20          this->doubles = newBuffer.doubles;
21      }
22
23      void print() const {
24          for (int i=0; i<size; ++i) cout << doubles[i] << " ";
25          cout << endl;
26      }
27 };
28
29 int main() {
30     Buffer b1(5); b1.print();
31     Buffer b2(5); b2.fill(12.34); b2.print();
32 }
```

כשמריצים את התוכנית מתקבל הפלט:

```
*** Error in a.out: double free or corruption (fasttop):
0x000000001178c90 ***
```

א [1 נק']. מה סוג השגיאה? (קומפילציה / קישור / זמן-ריצה / לוגית / דליפת-זיכרון).

ב [4 נק']. מה גורם לשגיאה? ציינו מספרי שורות והסבירו.

ג [5 נק']. איך אפשר לתקן את המחלקה Buffer כך שהתוכנית תרוץ בלי שגיאות?

שאלה 5 [10 נקודות]

נתונה התוכנית:

```
01  #include <iostream>
02  using namespace std;
03  class Board {
04      int size;
05      char* data;
06  public:
07      Board(int size) { this->size = size; data = new char[size]; }
08      Board(const Board& other) {
09          data = new char[other.size];
10          for (int i=0; i<other.size; ++i)
11              data[i] = other.data[i];
12      }
13      void fill(char c) {
14          for (int i=0; i<size; ++i) {
15              data[i] = c;
16          }
17      }
18  };
19
21  int main() {
22      Board b1(10);
23      b1.fill('x');
24      Board b2 = b1;
25      b2.fill('.'); // should not affect b1
26  }
```

כשמריצים make מתקבל הפלט:

```
make: *** [all] Segmentation fault (core dumped)
```

א [1 נק']. מה סוג השגיאה? (קומפילציה / קישור / זמן-ריצה / לוגית / דליפת-זיכרון).

ב [4 נק']. מה גורם לשגיאה? ציינו מספרי שורות והסבירו.

ג [5 נק']. איך אפשר לתקן את התוכנית כך שתרוץ בלי שגיאות?

שאלה 6 [40 נקודות]

נתונה התוכנית הראשית [ראו הסבר בעמוד הבא]:

```
#include <iostream>
using namespace std;

#include "AbstractTable.h"
#include "IntegerTable.h"
#include "StringTable.h"

int main() {
    cout << AbstractTable::count() << endl; // 0
    AbstractTable* tables[3];
    cout << AbstractTable::count() << endl; // 0
    // tables[0] = new AbstractTable(5);
    /* This line should create a compilation error:
       "allocating an object of abstract class type 'AbstractTable'"
    */
    tables[0] = new IntegerTable(5);
    cout << AbstractTable::count() << endl; // 1
    tables[1] = new StringTable(5);
    tables[2] = new IntegerTable(7);
    cout << AbstractTable::count() << endl; // 3
    for (AbstractTable* t: tables)
        t->print();
    /* Should print (for example):
       Integer table with values: 0 674657 1 141 -14341
       String table with values: . . . . .
       Integer table with values: 989 0 0 0 13241 1513554 454
    */
    delete tables[2];
    cout << AbstractTable::count() << endl; // 2
    delete tables[1];
    delete tables[0];
    cout << AbstractTable::count() << endl; // 0
}
```

התוכנית שייכת למערכת עיבוד נתונים כלשהי, שבה יש שני סוגי טבלאות: IntegerTable, StringTable. גודל של טבלה נקבע באופן דינמי בזמן ריצה. בטבלה מסוג IntegerTable, הערכים אינם מאוחזלים; בטבלה מסוג StringTable, כל הערכים מאוחזלים ל". (נקודה).

המערכת סופרת כמה טבלאות פעילות יש בכל רגע בתוכנית ומאפשרת לקרוא מספר זה ע"י השיטה
`AbstractTable::count`.

עליכם לכתוב את כל הקבצים הדרושים כך שהתוכנית הראשית המצורפת תתקמפל ותרוץ ותדפיס פלט נכון
כפי שמוסבר בגוף התכנית. כיתבו כל קובץ בדף נפרד.

יש להקפיד על כללי הנדסת תוכנה שנלמדו, בפרט: תיעוד, הסתרת מידע וניהול זיכרון.

שימו לב: אחת השורות בתוכנית הראשית מחוקה ע"י הערה. עליכם לוודא ששורה זו מחזירה שגיאת
קומפילציה, כלומר שהקומפיילר לא מאפשר ליצור טבלה מסוג `AbstractTable`.

שאלה 7 [10 נקודות]

- כיתבו פונקציה גנרית בשם minmax המקבלת שתי סדרות מטיפוס-נתונים כלשהו.
- אם הסדרה הראשונה ארוכה יותר מהשנייה - הפונקציה תזרוק חריגה "sequence 1 is longer".
 - אם הסדרה השנייה ארוכה יותר מהראשונה - הפונקציה תזרוק חריגה "sequence 2 is longer".
 - אם הסדרות באותו אורך - הפונקציה תשנה את שתי הסדרות באופן הבא:
 - הסדרה הראשונה תכיל, בכל מקום i, את הקטן מבין שני הערכים של שתי הסדרות במקום i.
 - הסדרה השנייה תכיל, בכל מקום i, את הגדול מבין שני הערכים של שתי הסדרות במקום i.

מצורפת תוכנית ראשית לדוגמה. כיתבו את התוכן של הקובץ minmax.h בלבד כך שהתוכנית הראשית תתקמפל ותיתן את התוצאות הנכונות. אין צורך לכתוב את הקובץ minmax.cpp.

```
#include "minmax.h"
#include <forward_list>
#include <iostream>
#include <vector>
using namespace std;
int main() {
    forward_list<int> list1 ({1,-2,3,-4,5});
    vector<int> vec1 ({-1,2,-3,4,-5});
    minmax(list1.begin(), list1.end(), vec1.begin(), vec1.end());
    for (int i: list1) cout << i << " "; cout << endl; // -1 -2 -3 -4 -5
    for (int i: vec1) cout << i << " "; cout << endl; // 1 2 3 4 5
    int arr1[] = {8,6,4,2,0};
    minmax(vec1.begin(), vec1.end(), arr1, arr1+5);
    for (int i: vec1) cout << i << " "; cout << endl; // 1 2 3 2 0
    for (int i: arr1) cout << i << " "; cout << endl; // 8 6 4 4 5
    minmax(vec1.begin(), vec1.end(), arr1, arr1+4);
    // exception: "sequence 1 is longer"
    minmax(vec1.begin(), vec1.end(), arr1, arr1+6);
    // exception: "sequence 2 is longer"
}
```