

# שפת C++ - תרגיל 1

תאריך הגשה:

הגשה מאוחרת (בהפחתת 10 נקודות):

## הנחיות חשובות:

א. בכל התרגילים יש לעמוד בהנחיות הגשת התרגילים וסגנון כתיבת הקוד. שני המסמכים נמצאים באתר הקורס - הניקוד יכלול גם עמידה בדרישות אלו.

ב. אין להגיש קבצים נוספים שאינם קשורים לתרגיל.

ג. עליכם לקמפל עם הדגלים Wextra -Wall -Wvla -c example.cpp -o example.o ולוודא שהתוכנית מתקמפלת ללא אזהרות, **תכנית שמתקמפלת עם אזהרות תגרור הורדה בציון התרגיל**. למשל, בכדי לקמפל קובץ מקור בשם example.cpp יש להריץ את הפקודה:

```
g++ -Wextra -Wall -Wvla -c example.cpp -o example.o
```

ד. עליכם לוודא שהתרגילים שלכם תקינים ועומדים **בכל דרישות הקימפול והריצה במחשבי בית הספר** מבוססי מעבדי bit-64 (מחשבי האקווריום, לוי, השרת river). חובה להריץ את התרגיל במחשבי בית הספר לפני ההגשה. (ניתן לוודא שהמחשב עליו אתם עובדים הנו בתצורת bit-64 באמצעות הפקודה "uname -a" ויודא כי הארכיטקטורה היא 64, למשל אם כתוב x86\_64).

ה. לאחר ההגשה, בדקו את הפלט המתקבל בקובץ ה-PDF שנוצר מה submission script בזמן ההגשה. באם ישנן שגיאות, תקנו אותן על מנת שלא לאבד נקודות.

ו. **בדיקת הקוד לפני ההגשה, גם על ידי קריאתו וגם על ידי כתיבת בדיקות אוטומטיות (tests) עברו היא אחריותכם**

חישבו על מקרי קצה לבדיקת הקוד, בדיקות אוטומטיות יהיו חלק מבדיקת הקוד לציון

**חלק מקבצי הבדיקה לדוגמה שמוספקים על ידי צוות הקורס נבדקים על ידי הבדיקה**

**האוטומטית בהגשה (submission script). שימוש בקבצי הבדיקה לדוגמה הוא**

**באחריותכם. במהלך הבדיקה הקוד שלכם ייבדק מול קלטים נוספים לשם מתן הציון.**

## משימת תכנות ראשונה - משחק בול - פגיעה (50 נק')

בתרגיל זה תממשו גרסה של המשחק בול - פגיעה (mastermind), בתרגיל זה המחשב מגריל אותיות ומבקש מהמשתמש לנחש אותן.

מהלך המשחק:

- המשתמש מריץ את התוכנית עם שני פרמטרים: הראשון מייצג את אורך המחרוזת שצריך לנחש (L) והשני מייצג את האות הגדולה ביותר (באלפבית) שיכולה להופיע במחרוזת (C), כל האותיות הינן lowercase).
- המחשב מגריל L אותיות בין 'a' ל 'C'. (השתמשו בקובץ erand.h שבאתר הקורס בכדי לקבל תווים אקראיים)
- מכאן בכל שלב המחשב מבקש מהמשתמש להכניס את הניחוש שלו (על התכנית להדפיס "Please enter a guess of <L> letters between 'a' and '<C>':\n", המשתמש מכניס L אותיות, והמחשב מחזיר לו כמה "בולים" (bulls) היו וכמה "פגיעות" (cows) היו (על התכנית להדפיס "You got <bulls> bulls and <cows> cows.\n").  
כאשר: "בול" מייצג אות שהמשתמש שם במקום שלה, ו"פגיעה" מייצגת אות שהמשתמש הכניס, קיימת במחרוזת שהוגרלה - אבל לא במקום הנכון.  
המשחק נגמר כאשר המשתמש ניחש את כל האותיות בסדר הנכון, במצב זה על התוכנית להדפיס "You got <L> bulls and 0 cows.\nIt took you <n> guesses.\n" כש <n> הוא מספר הניחושים שלקח למשתמש למצוא את המחרוזת (כולל האחרון) ו<L> הוא אורך המחרוזת.

### בחירת המחרוזת הרנדומלית

בכדי שנוכל לבדוק את הקוד שלכם, נא השתמשו בפונקציה randomCharacter שמוכרזת בקובץ erand.h המצורף. (אתם יכולים לצורך בדיקה לממש אותו בעצמכם, אין לצרף את הקובץ הזה או את המימוש של הפונקציה הזו להגשה, אנו נצרף אותו בבדיקה)

את הפתרון לסעיף זה הגישו בקובץ MasterMindPlay.cpp, והוסיפו ל makefile שלכם מטרה mastermindplay ליצירת קובץ הרצה בשם זה.

דוגמאות להרצה של התכנית (קלט מהמשתמש מסומן בכחול):

```
$ ./mastermindplay 4 d
Please enter a guess of 4 letters between 'a' and 'd':
abbc
You got 2 bulls and 1 cows.
Please enter a guess of 4 letters between 'a' and 'd':
abcd
You got 4 bulls and 0 cows.
It took you 2 guesses.
```

```
$ ./mastermindplay 5 c
Please enter a guess of 5 letters between 'a' and 'c':
abbca
You got 2 bulls and 1 cows.
Please enter a guess of 5 letters between 'a' and 'c':
```

acaaa

You got 3 bulls and 1 cows.

Please enter a guess of 5 letters between 'a' and 'c':

aaaba

You got 5 bulls and 0 cows.

It took you 3 guesses.

הערות:

- א. כאשר אות מופיעה מספר פעמים - כמות ה"בולים" וה"פגיעות" שלה לא יעלה על מספר הפעמים שהיא מופיעה במחרוזת הקלט ובמחרוזת שנבחרה ע"י המחשב (הקטן מביניהם).  
ז"א אות בקלט או במחרוזת שהמחשב הגריל לא יכולה להיספר גם כ"בול" וגם כ"פגיעה" (כלומר, כל אות בקלט לא יכולה להתאים ליותר מאות אחת במחרוזת המוגרלת, ולהפך).  
למשל: אם המחשב בחר את המחרוזת aaba והמשתמש הכניס baab ניחוש זה נספר כבול (ה'a' שמסומנת באדום) ושתי פגיעות (האותיות שמסומנות בכחול).
- ב. מומלץ לקרוא את החלק הבא של התרגיל לפני שמממשים את החלק הזה.
- ג. פרמטר ברירת מחדל - אם לא ניתן מספר, המחשב צריך להגריל מחרוזת באורך 4, אם לא ניתנה אות המחשב צריך להגריל מחרוזת בה האות המקסימלית היא 'd'.
- ד. בכל מקרה של פרמטרים לא תקינים (יותר משני פרמטרים, שני מספרים/שתי אותיות/תו שהוא לא מספר או אות קטנה/ אפס/ מספר שלילי וכדו') על התוכנית להדפיס הודעת שגיאה אינפורמטיבית ולצאת.
- ה. בכל מקרה של קלט לא תקין (אותיות מחוץ לטווח שהוגדר, תווים שאינם אותיות lowercase, מספר לא נכון של אותיות) על התוכנית שלכם להדפיס הודעה אינפורמטיבית מתאימה ולצאת.
- ו. בכל מקרה של שגיאה עליכם לצאת עם קוד שגיאה 1, בכל מקרה של ריצה שמסתיימת בניחוש נכון של המחרוזת עליכם לצאת עם קוד 0.

## משימת תכנות שניה – פתרון למשחק בול – פגיעה (50 נק')

בתרגיל זה תידרשו לממש את אותו המשחק מהסעיף הקודם – רק שהפעם המשתמש הוא זה שבוחר את האותיות, והמחשב צריך לנחש אותן.

עכשיו המשתמש והמחשב החליפו תפקידים, והמשחק נראה כך:

1. המשתמש מריץ את התכנית עם מספר ואות – כמות האותיות שהוא בחר (L) והאות המקסימלית (C)

המחשב יבקש מהמשתמש לבחור מחרוזת (התוכנית תדפיס "Please choose <L> letters in the range 'a' to '<C>':\n")

3. מכאן בכל שלב התכנית תדפיס L אותיות (עליכם להדפיס "My guess is: <guess>\n", כש <guess> – הניחוש הנוכחי) והמשתמש יחזיר לה כמה "בולים" היו וכמה "פגיעות" היו (מופרדים ברווח).

המשחק נגמר כאשר המחשב ניחש את כל האותיות בסדר הנכון, בשלב זה על התוכנית להדפיס "I did it in <n> guesses!\n" (כש <n> – מספר הניחושים שלקח למחשב לעלות על המחרוזת).

את הפתרון לסעיף זה הגישו בקובץ MasterMindSolver.cpp והוסיפו ל makefile שלכם מטרה mastermindsolver ליצירת קובץ הרצה בשם זה.

דוגמאות להרצות אפשריות (קלט מהמשתמש מסומן בכחול):

```
$ ./mastermindsolver 4 d
Please choose 4 letters in the range 'a' to 'd':
My guess is: abbc
2 1
My guess is: abcd
4 0
I did it in 2 guesses!
```

```
$ ./mastermindsolver 4 f
Please choose 4 letters in the range 'a' to 'f':
My guess is: abee
1 1
My guess is: acef
0 3
My guess is: ebfc
4 0
I did it in 3 guesses!
```

הערות:

א. מומלץ למחזר קוד מהסעיף הראשון. (אתם רשאים לחלק את הקוד שלכם לקבצים כראות עיניכם)

ב. תכנית שתחזור על ניחושים או תיתן ניחושים שהם בוודאות לא נכונים (בהתאם לתוצאות של ניחושים קודמים) תגרור הורדת נקודות.

ג. פרמטר ברירת מחדל – אם לא ניתן מספר, המחשב מניח שהמחרוזת באורך 4, אם לא ניתנה

אות המחשב צריך להניח מחרוזת בה האות המקסימלית היא d.

ד. בכל מקרה של פרמטרים לא תקינים (יותר משני פרמטרים, שני מספרים/שתי אותיות/תו שהוא לא מספר או אות קטנה/ אפס/ מספר שלילי וכדו') על התוכנית להדפיס הודעת שגיאה אינפורמטיבית ולצאת.

ה. בכל מקרה של קלט לא תקין (סכום שני המספרים גדול מ L, הקלט הוא לא שני מספרים מופרדים ברווח) על התוכנית להדפיס הודעה אינפורמטיבית מתאימה ולצאת.

ו. אם המשתמש טועה, והתכנית רואה שאין יותר ניחושים אפשריים (תשובות סותרות מהמשתמש, למשל 4 פגיעות עבור abcd ושלוש עבור dbca, 4 פגיעות על bbbb, או 0 ו 0 על abcd כשהטווח הוא a-d, וכו') – יש להדפיס הודעה אינפורמטיבית מתאימה ולצאת.

## ניהול זיכרון

בתרגיל זה (כמו ביתר התרגילים בקורס) תידרשו להשתמש בניהול זיכרון דינמי.

הנכם נדרשים לבדוק שניהול הזיכרון של התכנה שלכם תקין, זאת תעשו ע"י שימוש ב valgrind.

בכדי ש valgrind יוכל לעבוד ולהציג לכם שגיאות יש לקמפל את התכנה שלכם עם הדגל -g ולהריץ את התכנה באמצעות הפקודה

```
valgrind --leak-check=full --show-possibly-lost=yes --show-reachable=yes --undef-value-errors=yes <command>
```

ולוודא שהתכנה לא מתריעה על שגיאות.

## makefile

על קובץ ה makefile שלכם לתמוך בפקודות הבאות:

make mastermindplay

make mastermindsolver

ליצירת קבצי ההרצה עבור משימות התכנות.

make all

ליצירת שני הקבצים.

make clean

למחיקת קבצי ההרצה, הספריות וקבצים נוספים שיוצרו במהלך הבנייה (קבצי 0 למשל)

make tar

ליצירת קובץ ה tar להגשה.

## הגשה

עליכם להגיש קובץ בשם ex1.tar המכיל לפחות את הקבצים הבאים:

makefile MasterMindSolver.cpp MasterMindPlay.cpp

ניתן ליצור קובץ tar כנדרש ע"י הפקודה:

```
tar cvf ex1.tar <list of files to submit>
```

לפני ההגשה, פתחו את הקובץ ex1.tar בתיקיה נפרדת וודאו שהקבצים מתקמפלים ללא

שגיאות וללא אזהרות.

מומלץ גם להריץ בדיקות אוטומטיות וטסטרים שכתבתם על הקוד אותו אתם עומדים להגיש.

בתרגילים קודמים הגיעו אלינו מספר קבצים בפורמט שאינו tar, מה שהוביל לבעיות בבדיקה – נשמח אם תקפידו על הגשת קבצי tar.

בנוסף, אתם יכולים להריץ בעצמכם בדיקה אוטומטית עבור סגנון קידוד בעזרת הפקודה:

```
~labcpp/public/codingStyleCheck <file or directory>
```

כאשר <directory or file> מוחלף בשם הקובץ אותו אתם רוצים לבדוק או תיקייה שיבדקו כל הקבצים הנמצאים בה.

דאגו לבדוק לאחר ההגשה את קובץ הפלט (submission.pdf) וודאו שההגשה שלכם עוברת את ה-presubmission script ללא שגיאות או אזהרות.

**בהצלחה!**