

תרגיל בית 2 – באש, סקריפטים וטיפול בקבצים ב-C		
מועד ההגשה:	יום ב', 21/11/2022, בשעה 23:55	
האחראי על התרגיל:	פבל ליפשיץ	pavel@ee.technion.ac.il

הקדמה

במערכת ה-*grades* של הטכניון קרתה תקלה בתקופה הכי חשובה בסמסטר – תקופת מבחנים! בעקבות התקלה אין באפשרות המערכת לייצר היסטוגרמות ציונים וסטטיסטיקות עבור המבחנים השונים. הסטודנטים בקורס מבוא למערכות תוכנה בהנדסת חשמל החליטו לקחת אחריות ולעזור בתיקון של המערכת. בתרגיל זה נממש סט של כלים (כמו כלי Linux שאנו כבר מכירים, דוגמאות sort או uniq) אשר יקראו קבצי טקסט (או יקבלו אותם כקלט) עם נתונים על כל קורס והציונים בו, ונייצר היסטוגרמה וסטטיסטיקות נוספות הדרושות על ידי מערכת ה-*grades*. לשם פשטות התרגיל חולק עבורכם אך אין הכרח לעקוב אחר סדר ההוראות בו.

חלק ראשון – סקריפט באש לניהול המשימה

שם הסקריפט – *calc_stat.sh*

ארגומנטים – מספר קורס מהצורה *xxxxxx* (לדוגמא 044101).

על הסקריפט לבצע את הפעולות הבאות:

1. בדיקה שמספר הארגומנטים שהתקבלו לסקריפט תקין – במידה ולא תודפס הודעת השגיאה הבאה ל-STDERR: *Wrong number of arguments* ותבצע יציאה מסודרת מהסקריפט.
 2. בדיקה שקיים קובץ מהפורמט *xxxxxx.txt* בתיקייה – במידה ולא קיים קובץ כזה תודפס הודעת השגיאה הבאה ל-STDERR: *Course not found* ותבצע יציאה מסודרת מהסקריפט.
 3. יצירת תיקייה חדשה עבור תוצאות התכנית בפורמט הבא *xxxxxx_stat*, אם תיקייה כזו כבר קיימת יש למחוק אותה ואת התוכן שלה וליצור אחת חדשה ריקה.
- השתמשו בכלים שפיתחתם בחלק השני על מנת לייצר את התוכן הבא:
4. קריאת הציונים מקובץ הציונים *xxxxxx.txt* ויצירת היסטוגרמה של 10 bin-ים לקובץ חדש בשם *histogram.txt* תחת התיקייה שנוצרה בסעיף 3.
 5. יצירת קובץ חדש בשם *statistics.txt* שמכיל ציון ממוצע, ציון חציוני, הציון הנמוך ביותר והציון הגבוה ביותר בקורס מופרדים ב-TAB.
 6. השתמשו בכלי *hist* (מוסבר בהמשך) על מנת לחשב את אחוז הסטודנטים העוברים בקורס והוסיפו את התוצאה גם כן לקובץ *statistics.txt*.

חלק שני – תכניות c לחישוב הסטטיסטיקותשם התכנית – *median.exe*

על התכנית לבצע את הפעולות הבאות :

1. קבלת path לקובץ קלט כארגומנט ריצה לתוכנית או פעולה על stdin במידה ולא ציון שם קובץ כארגומנט, או לחלופין אם הארגומנט הראשון שווה ל- "-".
2. ביצוע הניתוח הבא :
- חישוב החציון של הקורס. לשם הבהרה אם ישנם N סטודנטים בקורס, החציון יהיה הציון שמיקומו במערך הינו $\frac{N+1}{2}$. $median_number$.
3. הוצאת הפלט ל-stdout.
4. במידה ויש ציון שאינו תקין (לא מספר בין 0-100) נוציא הודעה ל-stderr עם מספר השורה וטיב השגיאה.

שם התכנית – *mean.exe*

על התכנית לבצע את הפעולות הבאות :

1. קבלת path לקובץ קלט כארגומנט ריצה לתוכנית או פעולה על stdin במידה ולא ציון שם קובץ כארגומנט, או לחלופין אם הארגומנט הראשון שווה ל- "-".
2. ביצוע הניתוח הבא :
- חישוב הציון הממוצע לדיוק של 2 נקודות אחרי ה-0.
3. הוצאת הפלט ל-stdout.
4. במידה ויש ציון שאינו תקין (לא מספר בין 0-100) נוציא הודעה ל-stderr עם מספר השורה וטיב השגיאה.

שם התכניות – *min.exe, max.exe*

על התכנית לבצע את הפעולות הבאות :

1. קבלת path לקובץ קלט כארגומנט ריצה לתוכנית או פעולה על stdin במידה ולא ציון שם קובץ כארגומנט, או לחלופין אם הארגומנט הראשון שווה ל- "-".
2. ביצוע הניתוח הבא :
- חישוב הציון המירבי והציון הנמוך ביותר בהתאמה.
3. הוצאת הפלט ל-stdout.
4. במידה ויש ציון שאינו תקין (לא מספר בין 0-100) נוציא הודעה ל-stderr עם מספר השורה וטיב השגיאה.

שם התכנית – *hist.exe*

על התכנית לבצע את הפעולות הבאות:

1. קבלת path לקובץ קלט כארגומנט ריצה לתוכנית או פעולה על stdin במידה ולא צוין שם קובץ כארגומנט, או לחלופין אם הארגומנט הראשון שווה ל- "-".
2. קבלת מספר ה-bin-ים של ההיסטוגרמה כארגומנט (-n_bins). במידה ולא צוין, נבחר את n_bins להיות 10.
3. ביצוע הניתוח הבא:
 - חישוב ההיסטוגרמה של הקורס.
4. הוצאת הפלט ל-stdout לפי הפורמט הבא:

<מספר הציונים בטווח> TAB <טווח הבין>

דוגמא עבור n_bins=2 ו-86 ציונים תקינים:

0-49 13

50-100 73

5. במידה ויש ציון שאינו תקין (לא מספר בין 0-100) נוציא הודעה ל-stderr עם מספר השורה וטיב השגיאה.

פורמט קבצי הקלט

קבצי הקלט יכילו מספר רב של שורות, כאשר כל שורה תכיל ציון מספרי בטקסט. תוכלו להשתמש בפקודה הבאה על מנת לייצר מספר אקראי בטווח בין 0 ל-100 לצורך ייצור קלטים לבדיקת נכונות התוכנות שלכם:

```
echo "$(( $(od -An -N2 -i /dev/random) %100 ))"
```

בקובץ answers.txt ענו על השאלות הבאות:

1. כמה קוד "מחזרתם" בין כלי לבין כלי? האם ניתן היה לצמצם מחזור/חזרתיות זו?
2. (בנוסף 5 נק') היעזרו בכלים שפיתחתם וענו האם הציונים שהגרלתם בפקודה מפולגים יוניפורמית? במידה ולא הסבירו מדוע?

הוראות הגשה:

1. עברו היטב על הוראות ההגשה של תרגילי הבית המופיעים באתר טרם ההגשה! ודאו כי התכנית שלכם עומדת בדרישות הבאות:
 - א. התכנית קריאה וברורה.
 - ב. התכנית מתועדת היטב לפי דרישות התייעוד המופיעות באתר.
2. יש להגיש לינק ל-repository המכיל את הקבצים (שימו לב לשמות הקבצים עם case lower).
3. יש להגיש בפורמט הבא:


```
https://github.com/your-username/repository-name
123456789 student_1_mail@campus.technion.ac.il first_name_1 last_name_1
987654321 student_2_mail@campus.technion.ac.il first_name_2 last_name_2
```
4. שאלות בנוגע לתרגיל יש להפנות במייל **לאחראי על התרגיל בלבד**, ורק במידה והשאלה מכילה פתרון חלקי.
5. סיכום מפרט התרגיל:

סעיף	תיאור
נושא התרגיל	באש, סקריפטים וטיפול בקבצים ב-C
תאריך ההגשה	יום ב', 21/11/2022, בשעה 23:55
האחראי על התרגיל	פבל ליפשיץ pavel@ee.technion.ac.il
קבצי הקוד הנתונים	
קבצי הקלט והפלט הנתונים	
הקבצים שיש להגיש	<i>answers.txt</i> <i>calc_stat.sh</i> <i>max.c</i> <i>min.c</i> <i>mean.c</i> <i>median.c</i> <i>hist.c</i>

בהצלחה!

נספח – העברת פרמטרים לתכנית דרך פונקציית main

ההצהרה של פונקציית ה-main שלנו – `int main (int argc, char *argv[])`

- `argc` – פרמטר המכיל את כמות הפרמטרים שהתכנית קיבלה. ערכו תמיד לפחות 1 מאחר ושם התכנית גם הוא נחשב לפרמטר.

- `argv` – מערך של מחרוזות המכיל את הפרמטרים שה-main מקבל.

מערכת ההפעלה אחראית לעדכן את הערך של `argc` ולשחרר את מערך המחרוזות `argv`.

מצורפת דוגמא פשוטה לתכנית אשר מקבלת מספר משתנה של ארגומנטים משורת הפקודה ומדפיסה אותם:

```
main_arguments > C main_arg.c > main(int, char * [])
1  #include <stdlib.h>
2  #include <stdio.h>
3
4  int main (int argc, char *argv[]) {
5      printf("argc = %d\n",argc);
6      printf("argv[] content is:\n");
7      for (int i = 0; i < argc; i++) {
8          printf("arg[%d] = %s\n",i, argv[i]);
9      }
10
11     return 0;
12 }
```

```
File Actions Edit View Help
mamam@mamat-vm:~/HW2/main_arguments$ ./prog.exe
argc = 1
argv[] content is:
arg[0] = ./prog.exe
mamam@mamat-vm:~/HW2/main_arguments$ ./prog.exe hi hey
argc = 3
argv[] content is:
arg[0] = ./prog.exe
arg[1] = hi
arg[2] = hey
mamam@mamat-vm:~/HW2/main_arguments$ ./prog.exe hi hey 444
argc = 4
argv[] content is:
arg[0] = ./prog.exe
arg[1] = hi
arg[2] = hey
arg[3] = 444
mamam@mamat-vm:~/HW2/main_arguments$ ./prog.exe hi hey 444 "im here"
argc = 5
argv[] content is:
arg[0] = ./prog.exe
arg[1] = hi
arg[2] = hey
arg[3] = 444
arg[4] = im here
```

נספח – קימפול (Compile) ולינקוג' (Link)

קומפילציה של קובץ `c`. לקובץ `.o`.

נאמר שיש לנו קובץ `test.c` שנרצה לקמפל –

```
mamat@mamat-vm:~/Downloads/HW2/compile$ ls -l
total 4
-rw-rw-r-- 1 mamat mamat 194 Apr  7 11:10 test.c
```

נריץ את השורה הבאה בטרמינל – `gcc -c -g -Wall test.c`

```
mamat@mamat-vm:~/Downloads/HW2/compile$ gcc -c -g -Wall test.c
mamat@mamat-vm:~/Downloads/HW2/compile$ ls
test.c test.o
mamat@mamat-vm:~/Downloads/HW2/compile$
```

`gcc` – שם הקומפיילר שלנו

`-g` – דגל עבוד דיבאג

`-Wall` – דגל עבור אזהרות

כעת יש בידינו קובץ `.o`. ועלינו ללנקג' אותו לקובץ בר הרצה (`.exe`).

נריץ את השורה הבאה בטרמינל – `gcc -o prog.exe test.o`

```
mamat@mamat-vm:~/Downloads/HW2/compile$ gcc -o prog.exe test.o
mamat@mamat-vm:~/Downloads/HW2/compile$ ls
prog.exe test.c test.o
mamat@mamat-vm:~/Downloads/HW2/compile$
```

`gcc` – שם הלינקר שלנו

`prog.exe` – שם התכנית שאנו יוצרים

ניתן לאחד את שתי הפעולות לפעולה אחת על ידי הרצת הפקודה הבאה –

`gcc -g -Wall test.c -o prog.exe`

```
mamat@mamat-vm:~/Downloads/HW2/compile$ gcc -g -Wall test.c -o prog.exe
mamat@mamat-vm:~/Downloads/HW2/compile$ ls
prog.exe test.c
mamat@mamat-vm:~/Downloads/HW2/compile$
```