

## תרגיל בית 1

**מועד הגשה:** עד יום שני 8.11 בשעה 23:55. לא יהיו דחיות (עד כדי סיבות המאושרות ע"י תקנון האוניברסיטה).

### מטרת התרגיל

- ריענון השימוש בסביבת הפיתוח visual studio 2019.
- ריענון של תכנות בשפת C עם פעולות קלט/פלט והקצאה דינאמית של זיכרון.
- תרגול תכנות נכון: קוד קריא, חלוקה למודולים, שימוש בקבועים, תיעוד ועוד.
- היכרות ראשונית ובסיסית עם יצירת תהליכים ו-WinAPI.

### הגשה

צורת ההגשה מפורטת במסמך "הנחיות להגשת תרגילי בית – תשפ"ב" שבאתר המודל. אנה הקפידו למלא אחר ההוראות.

הגישו פרויקט מלא, כולל קבצי פרויקט (\*.\*sln, \*.vcxproj, \*.filters.vcxproj) של Studio Visual 2019, באופן שיאפשר לבדוק התרגילים לפתוח את הפרויקט על ידי לחיצה כפולה על קובץ solution-ו ולקמפל את הפרויקט ללא התראות או שגיאות.

הגישו בנוסף את תיקיית ה-Debug עם ה-Exe-ים.

### דגשים

הקפידו על קוד קריא ומתועד (ניקוד ירד על קוד לא מתועד כראוי).

בדקו את תקינות הקוד שכתבתם בסוף של סעיף.

אנו מעודדים אתכם לנסות תחילה לחפש תשובות באינטרנט (ד"ר גוגל), כאשר מדובר בשאלות תכנות כלליות. במקרים בהם גוגל לא הצליח לעזור, הפורום עומד לשירותכם.

ניתן להניח כי ההודעות יהיו באורך המתחלק ב-16 (הסיבה מתבררת בהמשך).

בתרגיל זה תתעסקו לא מעט בעבודה עם יצירת תהליכים, אך תהיה גם התעסקות עם קבצים.

בתרגיל זה נקרא ונכתוב לקבצים בתים שאינם בהכרח תווים קריאים, כדי לבדוק את הפלט שלכם לא תוכלו לקרוא את הקובץ הסופי ב-notepad כרגיל.

דוגמאות לפתרון:

- עם האפליקציה ++notepad ניתן לראות את כל הבתים האפשריים (לא בצורה הקסהדצימלית, אך יש טבלאות המסמלות מה כל בית מסמל).

אנשים המוצאים כלים ופתרונות חדשים מוזמנים לרשום על כך בפורום ולשתף את שאר הסטודנטים.

לפי המתואר לעיל, יכולה להיווצר בעיה בכתיבת בתים לקובץ וקריאת הבתים מהקובץ עם הפונקציות המוכרות fread, fwrite, ולכן כדאי להשתמש בפונקציות של API של windows (readfile, writefile) (ובדומה).

## סקירה כללית

מטרת התרגיל היא יצירת היכרות והבנה בסיסית של יצירת תהליכים (processes) לצד ריענון כתיבת קוד בשפת C.

הנחות והנחיות

- הימנעו משימוש במספרי קסם ומחרוזות מפורשות בקוד עצמו – רכזו את כל זה בקובץ ה- `HardCodedData.h` שיכיל את כל ה-`#define` ים שלכם.
- בכל מקרה שבו לא מוגדרת התנהגות נדרשת, אתם רשאים לבחור בכל התנהגות סבירה. התנהגות סבירה כוללת סיום אלגנטי של התוכנית ללא קריסה, והודעת פלט של שגיאה מתאימה.
- בכל הפרויקטים שניצור צריך להיות קובץ `main.c` המהווה את נקודת הכניסה אל התוכנית.
- זכרו לשחרר את הזיכרון הדינאמי שהקצתם במהלך התכנית.
- זכרו לסגור `HANDLE` ים.
- דאגו לחלק את הקוד שלכם בצורה ראויה לפי קונבנציות קוד בסיסיות. על פונקציות לא להיות ארוכות מדי, מסודרות ובנויות נכון (לכל פונקציה משמעות אחרת ללא שכפול קוד, עם מטרה אחת בלבד). בנוסף, על ה-`main` לא להיות ארוך מדי והוא ישמש בעיקר לקריאה לפונקציות אחרות והגדרת משתנים נחוצים.
- עבדו לפי הקונבנציות שהוגדרו במסמך `coding conventions` שבאתר המודל.

## רקע כללי

בתרגיל זה תכתבו שתי תכניות שונות, כלומר, התרגיל יכיל שני פרויקטים שונים, לכל אחד מהם יהיה קובץ `main.c` משלו וכל אחד מהם יקומפל לקובץ `exe`. נפרד. המטרה היא שתוכנית אחת תעשה שימוש בתוכנית השנייה.

לתהליך שמריץ את התוכנית שמריצה את התכנית השנייה נקרא תהליך האב, ולתהליך שהאב יוצר ומריץ את התכנית השנייה נקרא תהליך הבן.

הערה: התרגיל נכתב בלשון זכר אך פונה לאבות, אימהות, בנים ובנות.

## התרגיל

בתרגיל זה נממש מצפין בלוקים פשוט בהינתן מפתח והודעה להצפנה.

עוד על הצפנת בלוקים ניתן לקרוא [באן](#).

הסבר קצר על הצפנת בלוקים:

בהצפנת בלוקים יש הודעה באורך לא ידוע (שאותה נרצה להצפין), ומפתח באורך ידוע (במקרה שלנו המפתח יהיה 16 בתים).

על מנת להצפין את ההודעה המיועדת, נבצע XOR בין בתי המפתח לבתי ההודעה בצורה ציקלית – משמע, נקסר את 16 הבתים הראשונים של ההודעה ל-16 הבתים של המפתח, לאחר מכן נקסר את 16 הבתים הבאים לאחר מכן בהודעה לבתי המפתח כך שהבית ה-17 של ההודעה יקסר לבית הראשון של המפתח. ניתן להניח כי ההודעות יהיו באורך המתחלק ב-16.

## שלב 1

בתור התחלה פתחו solution חדש בשם Ex1, ובתוכו פרויקט בשם Son. בתוך פרויקט זה ממשו את התוכנית הנ"ל.

בסעיף זה נכתוב תכנית המקבלת כקלט דרך ה-command line שם של קובץ בו נמצאת ההודעה שאותה נרצה להצפין, offset של בתים בתוך ההודעה כדי לדעת מאיפה להתחיל לקרוא 16 בתים להצפנה, ושם של קובץ בו יהיו 16 בתי המפתח.

הקלטים יראו בצורה הבאה:

<plaintext file name> <offset> <key file name>

כאשר:

plaintext file name - שם הקובץ בו נמצאת ההודעה הכוללת אותה עלינו להצפין.

offset - ההיסט בביתים ממנו עלינו להתחיל לקרוא את ה-16 בתים שעלינו להצפין.

key file name - שם הקובץ בו נמצא המפתח איתו נצפין את ההודעה.

לדוגמה:

```
C:\Users>Son.exe ./plaintext.txt 16 ./key.txt
```

התכנית תקרא 16 בתים מהoffset המתאים בקובץ של ההודעה, ואת המפתח מהקובץ של המפתח, תבצע xor ביניהם, כאשר הבית הראשון של ההודעה מקוסר עם הבית הראשון של המפתח, הבית השני של ההודעה מקוסר לבית השני של המפתח, וכך הלאה.

דוגמה:

הודעה: Hi I'm a student

מפתח: oJ8Gre8B#&F2#BDK

נצטרך לבצע:

H ^ o

i ^ J

...

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
t	n	e	d	u	t	s	0x20	a	0x20	m	'	l	0x20	i	H
↕	↕	↕	↕												
K	D	B	#	2	F	&	#	B	8	e	r	G	8	J	o

הערה: 20x0 הוא ייצוג הקסדהצימלי של הבית של רווח לפי טבלת ASCII

לאחר פעולת ה-xor נכתוב את 16 הבתים של התוצאה לתוך קובץ הנקרא Encrypted\_message.txt (בצורה שלא תדרוס את המידע שנמצא בביתים אחרים לפני/אחרי 16 הבתים הנכתבים). כפי שתראו בהמשך, זהו קובץ שיכיל בסוף הריצה של תכנית האב את ההודעה המוצפנת הסופית במלואה, כאשר כל בן יוסיף את ה-16 בתים אותם הוא הצפין.

תכנית הבן תחזיר ב-exitcode אם הקוד סיים בהצלחה או לא. אם הקוד הסתיים בהצלחה ללא שגיאות, הקוד יחזיר את הערך 0. במקרה של בעיה בתוכנית (פתיחת קובץ לא הצליחה, שגיאה בכתיבה וכל באג אחר) תכנית הבן תסיים ללא קריסה ותחזיר ב-exitcode את המספר 1.

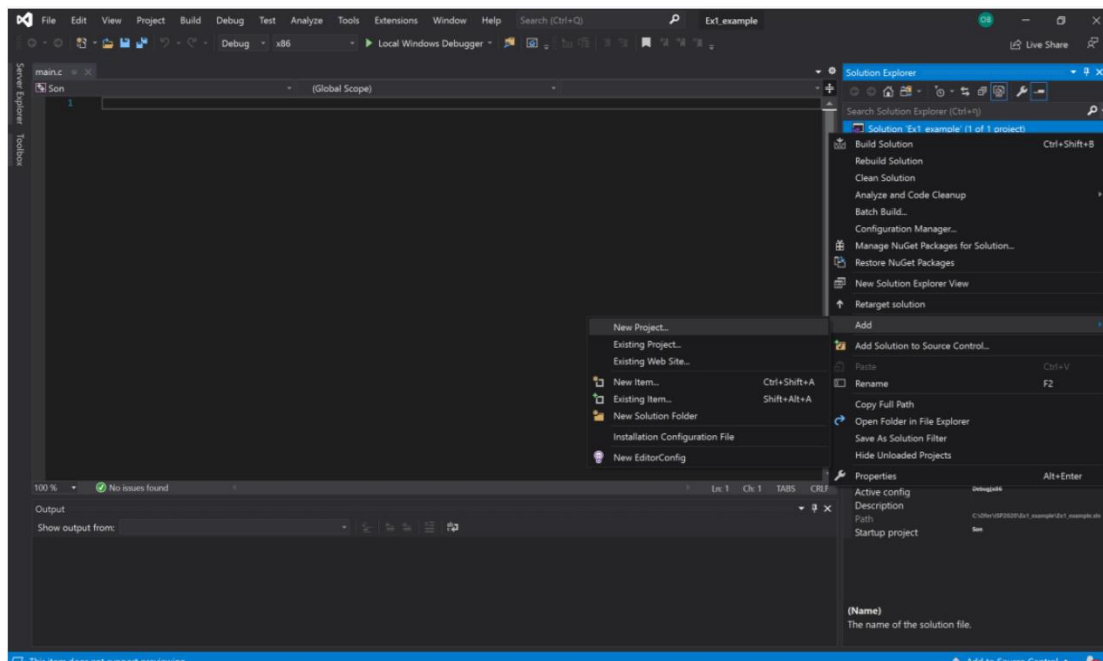
ניתן להניח קלט תקין לתכנית הבן.

טיפ: מתוך תכונות x86, כאשר תצפינו את ההודעה ולאחר מכן תבצעו את ההצפנה פעם נוספת על ההודעה המוצפנת, אתם אמורים לקבל את ההודעה המקורית. זו דרך פשוטה לבדוק את תקינות ההצפנה (אך לא כדאי להסתמך רק על שיטה זו, תשוו גם לקבצי קלט/פלט שקיבלתם במודל).

יש לבדוק את ערך החזרה של כל פונקציה שבה אתם משתמשים ועשויה להיכשל, ולהתמודד בהתאם להנחיות התרגיל.

## שלב 2

כעת נוסיף פרויקט נוסף ל-solution שלנו:



קראו לפרויקט החדש בשם Empty project (Father).

## שלב 3

בפרויקט החדש של האב כתבו תכנית שתקבל כקלט 2 שמות של קבצים.

בקובץ אחד תהיה ההודעה שעליכם להצפין, ובקובץ השני 16 בתים שישמשו כמפתח שלכם.

התוכנית תיצור עבור כל 16 בתים בהודעה הגלויה תכנית בן כאשר הבן יבצע את פעולת ה-xor בין 16 הבתים של ההודעה לבין המפתח (שימו לב לארגומנטים של תכנית הבן ואיך צריך לממש זאת).

על תכנית האב לחכות שתכנית הבן תסיים את הפעולה ואת הכתיבה לקובץ ולקבלת exitcode ראוי, ורק לאחר מכן ליצור תכנית בן חדשה.

יש להצפין את ההודעה בעזרת הבנים עם סדר כרונולוגי של ההודעה (זאת אומרת קודם כל יוצפנו בתים 0-15, לאחר מכן בתים 16-31 וכך הלאה).

יש צורך לבדוק את ה-exitcode של כל בן ולבדוק האם הבן סיים את הפעולה בהצלחה או לא.  
אם הבן נכשל, יש לפעול לפי ההנחיות ולסיים את התוכנית בצורה אלגנטית עם הודעת שגיאה מתאימה למסך.

הערה: ניתן להניח כי קובץ הפלט של ההודעה המוצפנת יהיה באותו אורך של ההודעה הגלויה.

חשוב : זכרו לקרוא את ההנחיות וההנחות בתחילת התרגיל!

**בהצלחה!**