



תרגיל בית 1

הנחיות:

1. **תאריך הגשה: 29.12.2022 בשעה 23:50**
2. **הגשה בזוגות בלבד.**
יש להירשם כקבוצה במודל לפני ההגשה.
רק אחד משני השותפים בקבוצה צריך להגיש את התרגיל.
3. **הורידו את קבצי השלד מהמודל וממשו את הפונקציות החסרות בהם.**
4. במודל נמצא לינק לאתר בו מופיעות הנחיות כתיבת קוד בפורמט של פרויקט GNU.
בנוסף מצורף קובץ PDF עם סיכום הנחיות הקידוד בעברית.
הקוד צריך להיכתב בהתאם למסמך ההנחיות. אנא קראו אותו בקפידה.
הגשה שלא תעמוד בהנחיות תגרור הורדת ניקוד.
5. הגשה של קובץ zip בודד. שם הקובץ בפורמט הבא:
HW1_ID1_ID2.zip
כאשר ID1 ו-ID2 הם תעודות הזהות של המגישים.
6. בתוך קובץ ה-zip נמצאים שני קבצים:
HW1Q1.c
HW2Q2.c
7. וודאו כי הקוד שלכם מתקמפל. קוד שאינו עובר קומפילציה יגרור הורדה משמעותית בציון.
8. בדקו את הקוד שלכם עם קלטים שונים ע"מ לבחון מקרים נוספים ומקרי קצה!
ע"מ לבצע בדיקות נוספות צרו את הקוד הנמצא בפונקציית ה-main (הוסיפו/שנו את הקלטים הקיימים) – מומלץ לעשות זאת בהעתק של הקובץ בגלל ההנחייה הבאה:
שימו לב: את ההגשה הסופית יש לעשות כאשר פונקציית ה-main זהה לחלוטין למצבה כאשר קיבלתם את הקובץ. אם בוצעו שינויים בפונקציה – החזירו אותה למצבה המקורי לפני ההגשה.
9. במקביל לפתיחת התרגיל נפתח פורום שאלות במודל.
שאלות לגבי התרגיל ניתן לשאול שם.



משימה 1

בתרגילי הבית הקרובים תממשו מספר תוכניות בשפת C שמטרתן לפתור בעיות בהן נתקלת חברת האוטובוס "טכנובוס", צוות ההנהלה ונהגי האוטובוסים שלה.
בתקנון חברת "טכנובוס" קיימים שני הסעיפים הבאים:

- א. הנהגים נדרשים לסייע ל"נוסעים מבוגרים" בעלייתם וירידתם מהאוטובוס. נהג יסייע לנוסע מבוגר רק אם הנוסע ביקש סיוע.
"נוסעים מבוגרים" מוגדרים כנוסעים שגילם 80 ומעלה.
- ב. ילדים מתחת לגיל 8 (כולל) לא נדרשים לשלם בעבור נסיעה.

אתם נדרשים לממש פונקציה בעלת החתימה הבאה הנמצאת בקובץ HW1Q1.c:

```
void handle_passengers(int bus_num, int age[], int assist[], int length)
```

הפונקציה מקבלת את מספר הקו, שני מערכים ואת גודל המערכים:

- מערך age מכיל את גיל הנוסעים
- מערך assist מכיל את המספר 1 באינדקס המתאים לנוסע אם הוא ביקש סיוע, אחרת 0

לנהגים אסור לסייע לאנשים שאינם מבוגרים. לכן אם נוסע שאינו מבוגר ביקש סיוע, אין לספק לו סיוע.
בתחילת הנסיעה תודפס הודעה עם מספר הקו ואיחול נסיעה נעימה.

עליכם לכתוב קוד המבצע את הפעולות הבאות:

עליכם להדפיס למסך הודעות לפי ההוראות הבאות (בהמשך המסמך מופיעות דוגמאות להדפסות):

1. במידה וניתן סיוע לנוסע מבוגר, עליכם להדפיס את גילו ואת האינדקס (המיקום) שלו במערך.
2. במידה ונוסע שאינו מבוגר ביקש סיוע, יש להדפיס הודעה שהוא לא בגיל המתאים לקבלת סיוע.
3. במידה וגיל הנוסע הוא 8 ומטה, עליכם להדפיס את האינדקס שלו, לברך אותו בנסיעה נעימה ולהודיע לו שהוא נוסע בחינם.

דוגמא:

עבור המערכים:

```
int age[] = {24, 32, 80, 90, 8, 9};  
int assist[] = {1, 0, 0, 1, 0, 1};  
int length = 6;  
int bus_num = 19;
```

יודפסו למסך ההודעות הבאות:

```
### Welcome to bus line 19! Have a nice ride! ###
```

```
I'm sorry! 24 is not old enough to receive assistant.  
Passenger 4 is 90 years old, allow me to help you.  
Welcome passenger 5! You ride free of charge!  
I'm sorry! 9 is not old enough to receive assistant.
```



ועבור המערכים:

```
int age[]    = {1, 2, 54, 89};  
int assist[] = {0, 1, 0, 1};  
int length  = 4;  
int bus_num  = 8745;
```

יודפסו למסך ההודעות הבאות:

```
### Welcome to bus line 8745! Have a nice ride! ###  
  
Welcome passenger 1! You ride free of charge!  
I'm sorry! 2 is not old enough to receive assistant.  
Welcome passenger 2! You ride free of charge!  
Passenger 4 is 89 years old, allow me to help you.
```

הערות:

- ניתן להניח שהקלט תקין, ששני המערכים בגודל זהה ושקיבלתם את האורך התקין של המערכים.
- הדפסת המשפט הראשון נתונה בקובץ המצורף, וודאו כי הדפסת יתר המשפטים מתבצעת בהתאם לפורמט הנתון בדוגמא (ירידות שורה, רווחים וכו').
- מצורף לתרגיל קובץ הפלט HW1Q1_output.txt, השתמשו בקובץ זה על מנת להשוות את התוצאות מהריצה שלכם, באופן ידני או בעזרת תוכנות השוואת תוכן (למשל DiffMerge).



משימה 2

מחשב יודע בפועל לעבוד עם ביטים (bit) בלבד, כאשר כל ביט הוא 0 או 1.
ע"מ לייצג דברים שאינם 0 או 1, המעבד מקבץ מספר ביטים יחד ליחידת זיכרון הנקראת בית (byte).
בצורה זו ניתן להציג מספרים גדולים מ-0 או 1 וביטויים שאינם מספרים.
למשל, הטיפוס Integer משתמש ב-32 ביטים (שהם 8 בתים), וניתן לשמור בעזרתו מספרים שלמים בתחום של כ- $\pm 2,000,000,000$ (כשני מיליארד).
ע"מ לייצג תו בודד, המחשב משתמש ב-8 ביטים (שהם בית יחיד).
המחשב מתרגם את הערך המספרי לתו בעזרת טבלה הנקראת ASCII Table (מצורפת בהמשך).
זוהי למעשה רשימה של 255 תווים, כאשר לכל תו מתאים מספר בין 0 ל-255.
כך שלמעשה, לכל תו במחשב יש מספר מזהה ייחודי, וכך המחשב מדפיס את התו המתאים.
לדוגמא, הערך 65 מתאים לתו 'A' (התבוננו בטבלה בתחתית הדף).
ניתן לראות זאת ב-C ע"י הדוגמא הבאה:

```
#include <stdio.h>
void main(){
    char character_example = 'A';
    int integer_example = 65;
    printf("printing character as integer: %d\n", character_example);
    printf("printing integer as character: %c\n", integer_example);
}
```

הרצת הקוד יוצרת את הפלט הבא:

```
printing character as integer: 65
printing integer as character: A
```

כלומר, השמה של תו (char) לתוך מספר שלם (int) היא פעולה חוקית ב-C.
קיימים גם תווים עבור מספרים – שימו לב שהתו אינו זהה לערך המספרי המייצג אותו!

- עבור התו '0', מתאים הערך המספרי 48
 - עבור התו '1', מתאים הערך המספרי 49
 - עבור התו '2', מתאים הערך המספרי 50
- וכך הלאה, עד התו '9', והערך 57.

הטבלה הבאה מכילה את רוב התווים הניתנים להדפסה מטבלת ה-ASCII:

Code	Char	Code	Char	Code	Char	Code	Char	Code	Char	Code	Char
32	[space]	48	0	64	@	80	P	96	`	112	p
33	!	49	1	65	A	81	Q	97	a	113	q
34	"	50	2	66	B	82	R	98	b	114	r
35	#	51	3	67	C	83	S	99	c	115	s
36	\$	52	4	68	D	84	T	100	d	116	t
37	%	53	5	69	E	85	U	101	e	117	u
38	&	54	6	70	F	86	V	102	f	118	v
39	'	55	7	71	G	87	W	103	g	119	w
40	(56	8	72	H	88	X	104	h	120	x
41)	57	9	73	I	89	Y	105	i	121	y
42	*	58	:	74	J	90	Z	106	j	122	z
43	+	59	;	75	K	91	[107	k	123	{
44	,	60	<	76	L	92	\	108	l	124	
45	-	61	=	77	M	93]	109	m	125	}
46	.	62	>	78	N	94	^	110	n	126	~
47	/	63	?	79	O	95	_	111	o	127	[backspace]

התבוננו בקטע הקוד ונסו להבין את ערכו של המשתנים הבאים מתוך הטבלה (התשובה בדף הבא):

```
int i = 'w', j = '@';
char k = 63, l = 32;
```



- התו 'w' מתאים לערך 119, לכן ערך i הוא 119.
- התו '@' מתאים לערך 64, לכן ערך j הוא 64.
- הערך 79 מתאים לתו '?', לכן k הוא התו '?'.
- הערך 32 מתאים לתו ' ' (space), לכן l הוא תו הרווח (המקש הארוך במקלדת).

מנהלי חברת "טכנובוס" חושדים כי קיים סוכן חשאי שהסתכן לחברה. המנהלים מאמינים כי הסוכן מצא דרך יצירתית להעביר מסרים סודיים למפעילים שלו בעזרת המספרים של קווי האוטובוס השונים וטבלת ה-ASCII.

הסוכן פרץ למערכת ניהול האוטובוסים של החברה והוא יכול להחליט איזה אוטובוס עובר באיזה תחנה ומתי כרצונו.

הסוכן בוחר בכל יום מספר מסוים, ודואג שקו האוטובוס הראשון בתחנה מסוימת מתאים למספר שבחר. לאחר מכן, מספרו של כל קו שעובר בתחנה במשך היום מתאים לתו מטבלת ה-ASCII, אך התו "מוזז" במספר של קו האוטובוס הראשון לאותו יום.

למשל, אם הסוכן בחר את המספר 17, וברצונו להביע את האות 'A', הוא ידאג שהאוטובוס הראשון שיעבור בתחנה הוא קו מספר 17.

אחריו יעבור הקו שמספרו מתאים לערך התו 'A' בטבלת ה-ASCII פחות 17, הערך המתאים לתו 'A' הוא 65,

לכן יעבור בתחנה קו מספר $65 - 17 = 48$.

דוגמא נוספת, אם הסוכן רוצה להעביר את המילה "Secret", ובוחר לבצע הזזה של 9, איזה קווי אוטובוס יצטרך לבחור?

נבדוק מה הם הערכים של התווים המתאימים למילה:

S	e	c	r	e	t
83	101	99	114	101	116

נחסר 9 מכל מספר, ונגלה את קווי האוטובוס:

$$(83 \ 101 \ 99 \ 114 \ 101 \ 116) - 9 = (74 \ 92 \ 90 \ 105 \ 92 \ 107)$$

ולכן, עבור המילה Secret והזזה ב-9, האוטובוסים שיעברו בתחנה באותו יום הם (זכרו שהאוטובוס הראשון הוא ערך ההזזה והוא מגיע כחלק מהמערך לפונקציה):

9 74 92 90 105 92 107

עליכם לכתוב קוד המפענח את המסרים הסודיים של הסוכן.

אתם נדרשים לממש פונקציה בעלת החתימה הבאה הנמצאת בקובץ HW1Q2.c:

```
void decipher(int buses[], int length)
```

הפונקציה מקבלת רשימה של קווי אוטובוס שעברו בתחנה החשודה, ואת אורך הרשימה. עליכם להדפיס את הקוד המפוענח.

הערות:

- ניתן להניח כי הקלט תקין – המערך מכיל מספרים, שלאחר ההזזה – הם תווי ASCII בטווח תקין, ושהמשתנה length אכן שומר את גודל המערך.
- מצורף לתרגיל קובץ הפלט HW1Q2_output.txt, השתמשו בקובץ זה על מנת להשוות את התוצאות מהריצה שלכם, באופן ידני או בעזרת תוכנות השוואת תוכן (למשל DiffMerge).

בהצלחה!