3.1 [2 points] Typescript for compilation

```
$ make clean
```

rm *.hex *.ihx *.lnk *.lst *.map *.mem *.rel *.rst *.sym
process_begin: CreateProcess(NULL, rm *.hex *.ihx *.lnk *.lst *.map *.mem *.rel *.rst *.sym, ...) failed.
make (e=2): 系統找不到指定的檔案。
make: *** [Makefile:25: clean] Error 2

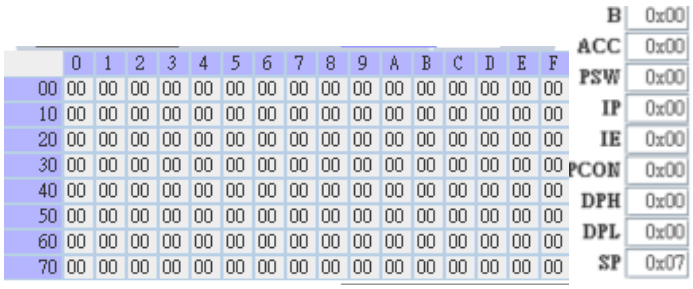
```
$ make
```

sdcc -c testcoop.c
testcoop.c:55: warning 158: overflow in implicit constant conversion
sdcc -c cooperative.c
cooperative.c:189: warning 85: in function ThreadCreate unreferenced function argument : 'fp'
cooperative.c:237: warning 158: overflow in implicit constant conversion
sdcc -o testcoop.hex testcoop.rel cooperative.rel

3.2 [18 points] Screenshots and explanation

- Take one screenshot before each `ThreadCreate` call. Explain how the stack changes.

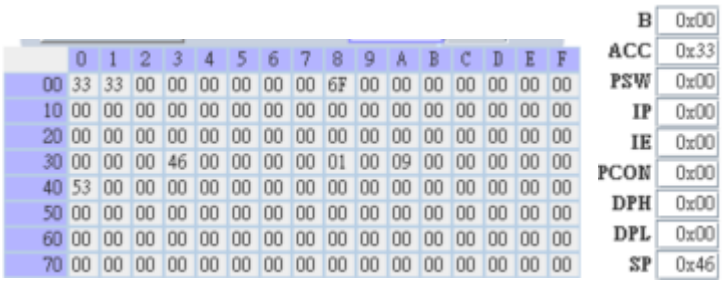
before ThreadCreate(main)



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
10	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
30	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
40	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
50	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
70	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

B	0x00
ACC	0x00
PSW	0x00
IP	0x00
IE	0x00
PCON	0x00
DPH	0x00
DPL	0x00
SP	0x07

after ThreadCreate(main)



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	33	33	00	00	00	00	00	00	6F	00	00	00	00	00	00	00
10	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
30	00	00	00	46	00	00	00	00	01	00	09	00	00	00	00	00
40	53	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
50	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
70	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

B	0x00
ACC	0x33
PSW	0x00
IP	0x00
IE	0x00
PCON	0x00
DPH	0x00
DPL	0x00
SP	0x46

```
C:   00000053  _main                                    testcoop
```

40 is set to 53 and SP is 0x46, so after RESTORESTATE is done, 53 will be pop and we will start to run on main. 3A, which is the temp that store the original SP is set to 09. So, after creating is done, SP can be set back to 09.

before ThreadCreate(Producer)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	33	33	00	00	00	00	00	00	6F	00	00	00	00	00	00	00
10	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
30	00	00	00	46	00	00	00	00	01	00	09	00	00	00	00	00
40	53	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
50	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
70	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

B	0x00
ACC	0x00
PSW	0x00
IP	0x00
IE	0x00
CON	0x00
DPH	0x00
DPL	0x09
SP	0x3F

```
C:   00000009  _Producer                                testcoop
```

DPL is set to 09 because it's calling ThreadCreate(Producer).

after ThreadCreate(Producer)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	33	33	00	00	00	00	00	10	6F	00	00	00	00	00	00	00
10	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
30	00	00	00	46	00	00	00	00	03	00	41	01	00	00	00	00
40	5C	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
50	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
70	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

B	0x00
ACC	0x4F
PSW	0x01
IP	0x00
IE	0x00
PCON	0x00
DPH	0x00
DPL	0x09
SP	0x4F

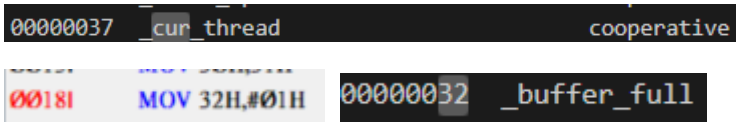
```
ØØ59I      LCALL ØØ84H
ØØ5CI      LJMP ØØ2DH
```

40 is set to 5C because 5C is the next instruction after ThreadCreate(Producer), SP is 0x4F because the new thread is found. 3A, which is the temp that store the original SP is set to 41. So, after creating is done, SP can be set back to 41. And 38 which is mask, is set to 03, indicating that the bit map is now 0011.

```
00000038  _mask                                         cooperative
```

- Take one screenshot when the `Producer` is running. How do you know?

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	33	33	00	00	00	00	00	01	33	34	00	00	03	00	00	02
10	00	33	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
30	45	45	00	46	56	00	00	01	03	00	41	01	00	00	00	00
40	4D	00	01	00	01	00	09	00	00	00	00	00	00	00	00	00
50	13	00	01	00	00	00	09	00	00	00	00	00	00	00	00	00
60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
70	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00



By observing the 37, which is set to 01, I can tell that the current thread is now 1, which is producer. And this line indicates setting buffer full to 1.

- Take one screenshot when the `Consumer` is running. How do you know?

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	33	33	00	00	00	00	00	01	33	35	00	00	03	00	00	04
10	00	33	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
30	46	47	01	46	56	00	00	00	03	00	41	01	00	00	00	00
40	4D	00	01	00	01	00	09	00	00	00	00	00	00	00	00	00
50	13	00	01	00	00	00	09	00	00	00	00	00	00	00	00	00
60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
70	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Observing the 37, which is set to 00, I can tell that the current thread is now 0, which is `Consumer`.