Assembly Assignment

This a group coursework. Form groups of four.

To complete the assignment, you will need to: -

- First work through the laboratory AVR studio.
- Refer to AT mega 8535 Data Sheet.
- Refer to AT mega 8535 Instruction Set.

Technical Support

For technical support contact CMS Technical Support

cms-support@gre.ac.uk

All deliverables must be completed, if any are not undertaken, your work will be marked to zero.

- A viva (16/12/2021),
- Your code, to be uploaded into the system on the 15/12/2020 with the following header:-

	Registration Number	Surname	Forename	% Contribution
Student 1	001131628	Chavush	Chisel	25%
Student 2	001141646	Iz	Ekrem Said	25%
Student 3	001174434	Turker	Selin	25%
Student 4	001141387	Hasan	Zahid	25%

```
.equ DDRA =$1A
.equ DDRB =$17
.equ PORTA =$1B
.equ PORTB =$18
ldi r20,$FF
out DDRA,r20 ;we set the port a for output with 0xFF
out DDRB,r20
ldi r21,$06 ;main loop counter 6 times
ldi r16,$66 ;we provide the seed number
mainloop:
mov r17,r16 ;we copy the seed number 2 times in order to play on them
mov r18,r16

andi r17,$01 ;we AND the copied seed number with 0x01 and 0x02 to keep the 0th and 1st bits
andi r18,$02
lsr r18 ;we shift the register that we AND with 0x02 to right in oder to make the bit we want to keep move on to first bit
eor r17,r18  ;we XOR the last 2 bits
lsr r17 ;whe shift the last bits to the carry flag and put it on most significant bit by rotating right
ror r17
lsr r16
or r16,r17
```

```
cpi r16,$5A  ;we compare the new random number with the 0x5A, which is one value incremented from 0x59

brcc mainloop ;if the carry flag is set to 0, jump back to mainloop. Otherwise keep going

mov r17,r16 ;we copy the new genarated number 2 times to out put the most and least significant digits

mov r18,r16

andi r17,$0F  ;we AND the two registers with 0x0F and 0xF0 to keep last and first nibbles.

andi r18,$F0

swap r18 ;we swap the nibbles to show most significant digits as it was least significant digit

out PORTB,r17 ;we output the digits that are stored in register 17 and register 18

out PORTA,r18

dec r21 ;we decrement the loop counter by 1

brne mainloop ;if the loop counter doesnt hit 0, it jumps back to the mainloop to create another random number

end: rjmp end ;loop forever
```