

**THE 8051 ASSEMBLY LANGUAGE PROGRAMMING**

1. Find the square of the contents of R5 and store the result in R0 (high byte), and R1 (low byte).

```
MOV  A, R5
MOV  0F0H, R5
MUL  AB
MOV  R0, 0F0H
MOV  R1, A
```
2. Set the stack pointer to 30H and push the contents of code memory at address 0070H to the top of stack.

```
MOV  81H, #30H
CLR  A
MOV  DPTR, #70H
MOVC A, @A+DPTR
PUSH ACC
```
3. Store the contents of RAM location 20H at the address contained in RAM location 08H.

```
MOV  R0, 08H
MOV  @R0, 20H
```
4. Subtract the contents of RAM location 13H from RAM location 2BH; put the result in RAM location 3CH.

```
MOV  A, 2BH
CLR  C
SUBB A, 13H
MOV  3CH, A
```
5. Increment the contents of RAM location 13H, 14H and 15H using indirect addressing.

```
MOV  R0, #13H
INC  @R0
INC  R0
INC  @R0
INC  R0
INC  @R0
```
6. Decrement the contents of internal ROM location 0123H and store the result in external RAM location 01BDH.

```
MOV  DPTR, #0123H
CLR  A
MOVC A, @A+DPTR
DEC  A
MOV  DPTR, #01BDH
MOVX @DPTR, A
```
7. Clear bit 3 of RAM location 30H without affecting any other bits.

```
MOV  A, #11110111B
ANL  A, 30H
MOV  30H, A
```
8. Set the carry flag to one if the number in A is even; set the carry flag to zero if the number in A is odd.

```
RRC  A
CPL  C
```

**(End)**