III. STRUCTURED ASSEMBLY LANGUAGE PROGRAMMING TECHNIQUES

Structured Data Types

Sets

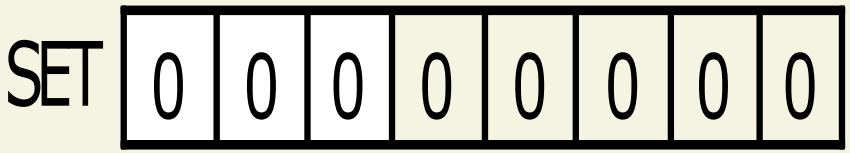
Set Operations:

- Add element
- Remove element
- Is element

- logic instructions/bitwise operations are used to implement set operations

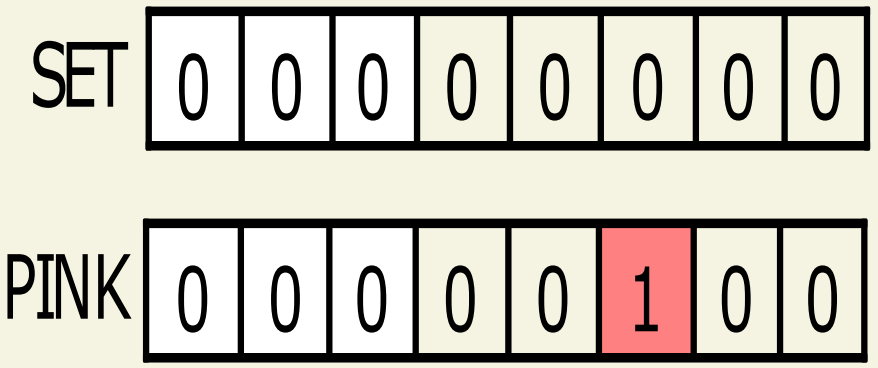
Sets – Add Element

```
BLUE      equ 1
GREEN     equ 2
PINK      equ 4
YELLOW    equ 8
ORANGE    equ 16
SET       db 0
```



Sets – Add Element

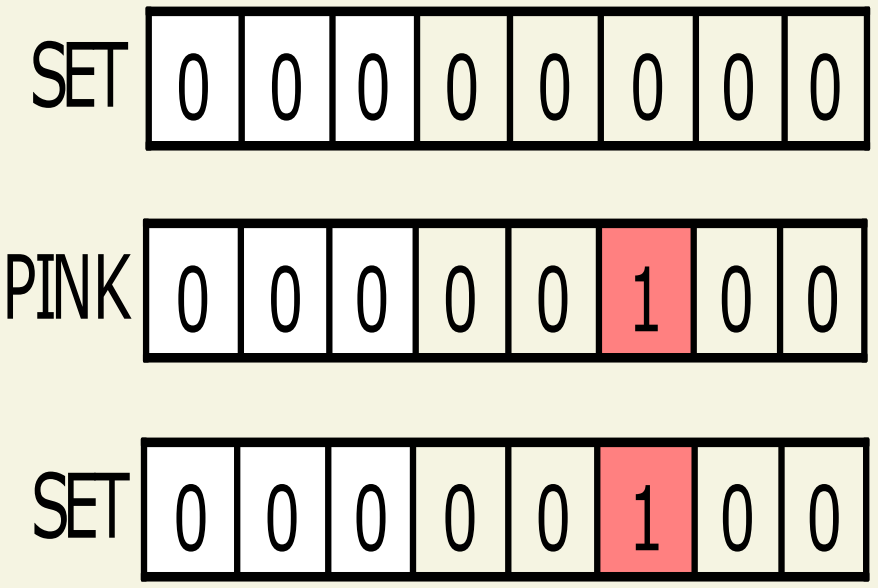
```
BLUE        equ 1
GREEN       equ 2
PINK        equ 4
YELLOW      equ 8
ORANGE      equ 16
SET         db 0
```



OR byte[SET], PINK

Sets – Add Element

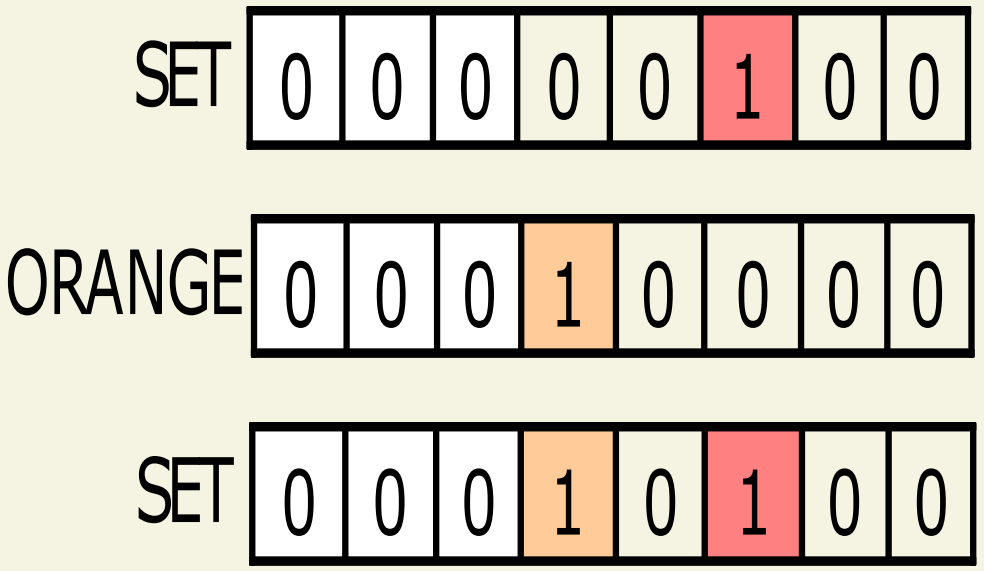
```
BLUE        equ 1
GREEN       equ 2
PINK        equ 4
YELLOW      equ 8
ORANGE      equ 16
SET         db 0
```



OR byte[SET], PINK

Sets – Add Element

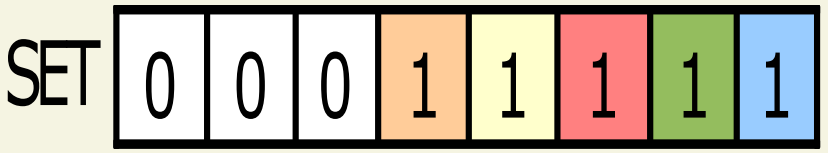
```
BLUE      equ 1
GREEN     equ 2
PINK      equ 4
YELLOW    equ 8
ORANGE    equ 16
SET       db 0
```




```
OR byte[SET], PINK
OR byte[SET], ORANGE
```

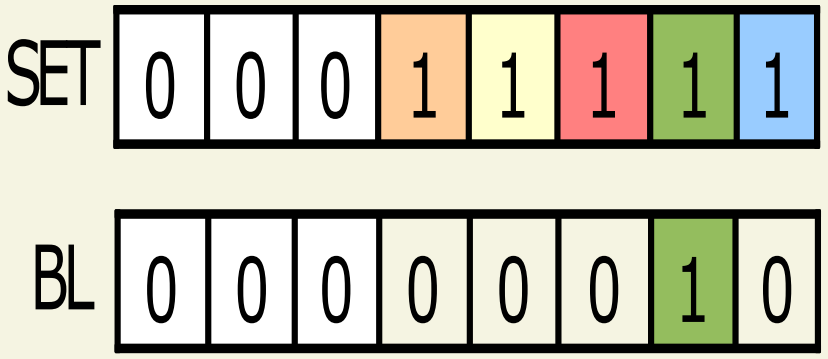
Sets – Remove Element

```
BLUE        equ 1
GREEN       equ 2
PINK        equ 4
YELLOW      equ 8
ORANGE      equ 16
SET         db 0
```



Sets – Remove Element

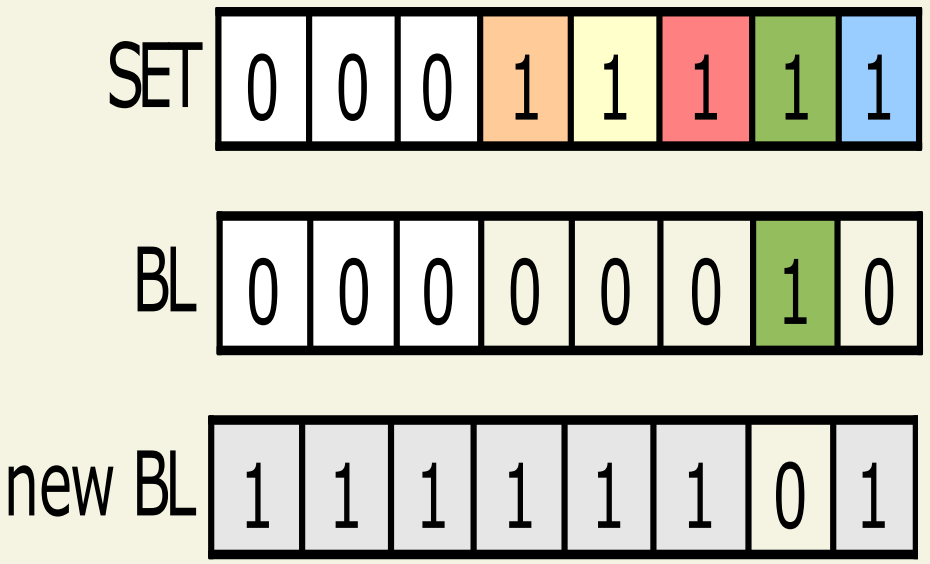
```
BLUE      equ 1
GREEN     equ 2
PINK      equ 4
YELLOW    equ 8
ORANGE    equ 16
SET       db 0
```



MOV BL, GREEN

Sets – Remove Element

```
BLUE        equ 1
GREEN       equ 2
PINK        equ 4
YELLOW      equ 8
ORANGE      equ 16
SET         db 0
```

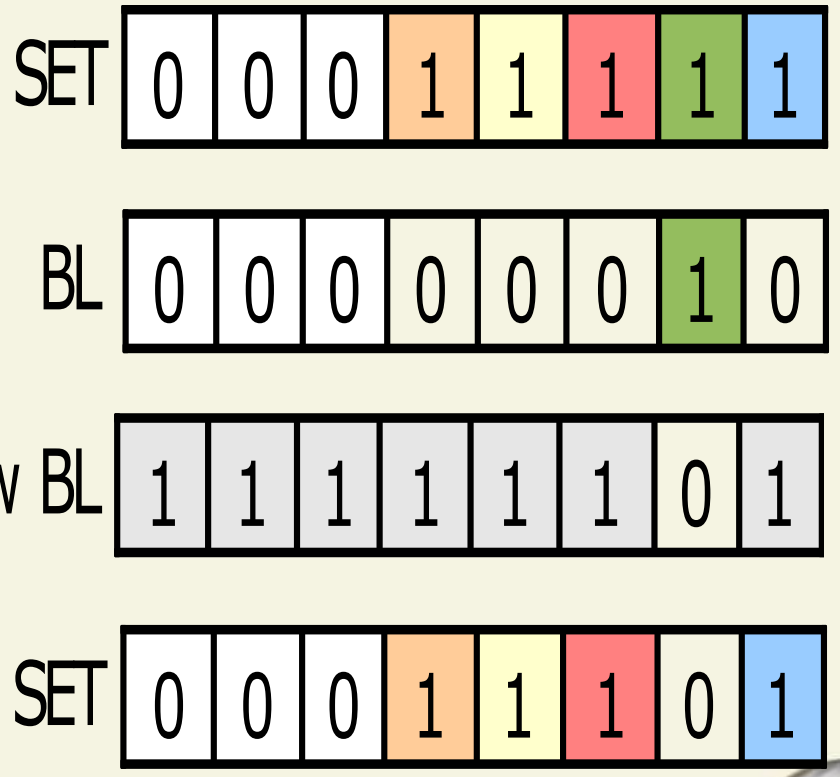



```
MOV BL, GREEN
NOT BL
AND byte[SET], BL
```

Sets – Remove Element

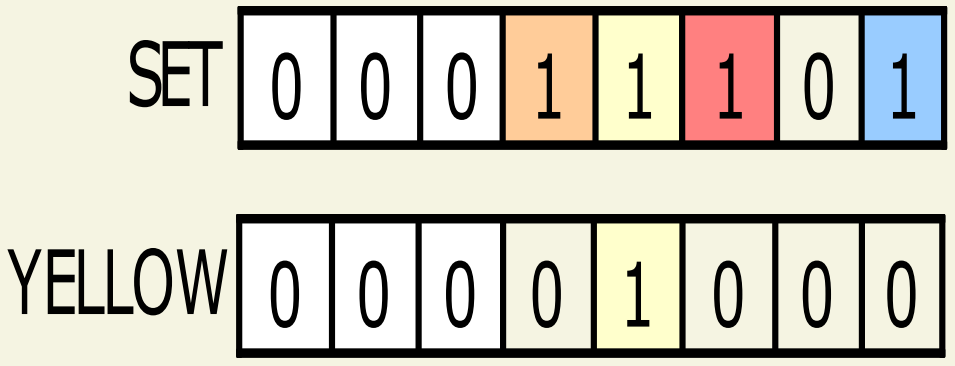
```
BLUE        equ 1
GREEN       equ 2
PINK        equ 4
YELLOW      equ 8
ORANGE      equ 16
SET         db 0

MOV BL, GREEN
NOT BL
AND byte[SET], BL
```



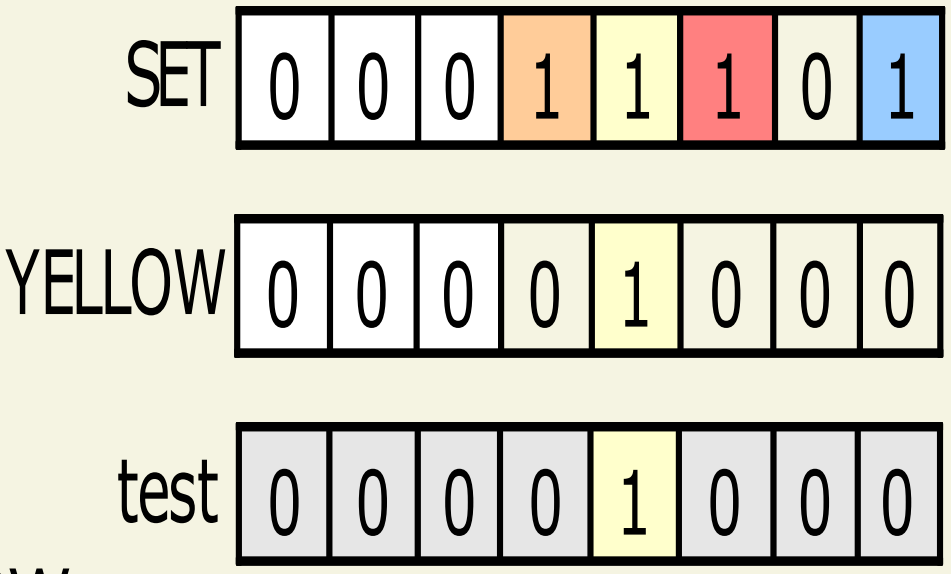
Sets – Is Element

```
BLUE        equ 1
GREEN       equ 2
PINK        equ 4
YELLOW      equ 8
ORANGE      equ 16
SET         db 0
```



Sets – Is Element

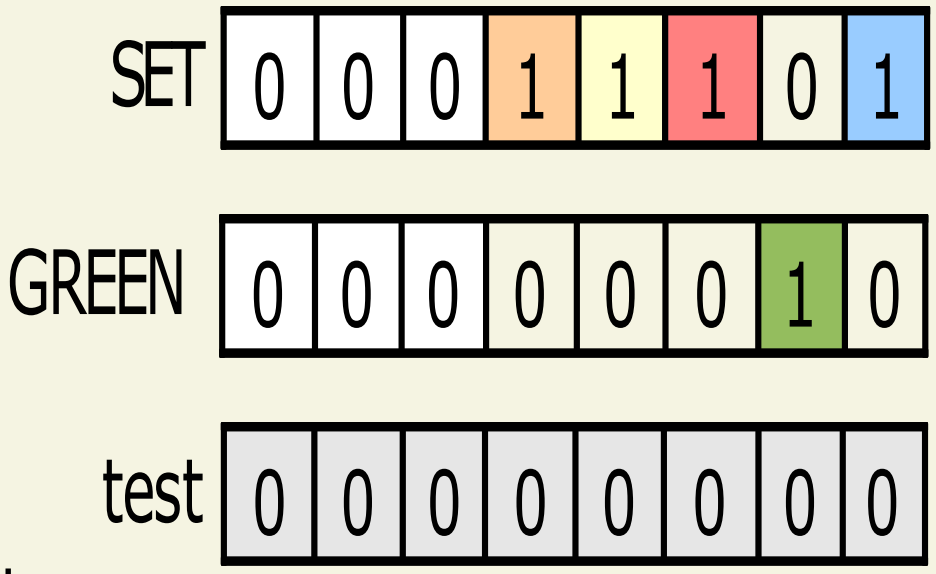
```
BLUE        equ 1
GREEN       equ 2
PINK        equ 4
YELLOW      equ 8
ORANGE      equ 16
SET         db 0
```




```
TEST byte[SET], YELLOW
JZ noYellow
```

Sets – Is Element

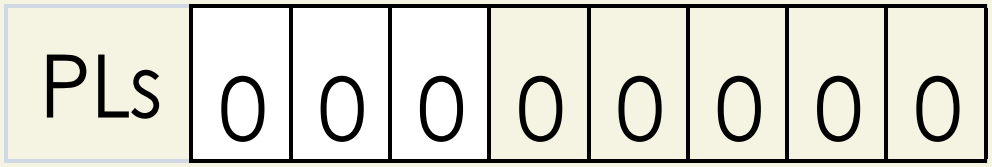
```
BLUE        equ 1
GREEN       equ 2
PINK        equ 4
YELLOW      equ 8
ORANGE      equ 16
SET         db 0
```




```
TEST byte[SET], GREEN
JZ noGreen
```

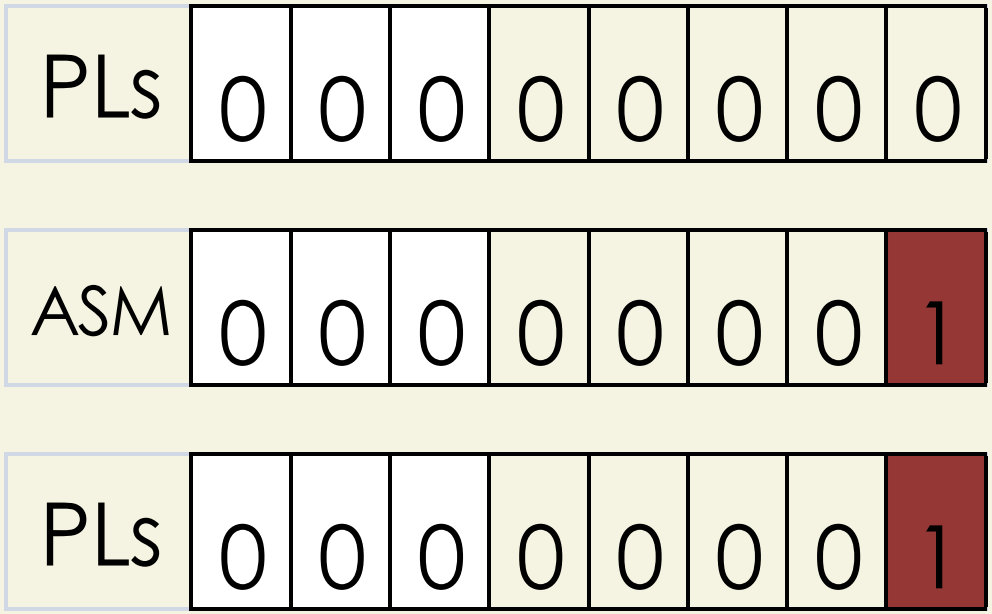
Sets – Add Element

```
ASM     equ 1
C#      equ 2
COBOL   equ 4
JAVA    equ 8
LISP    equ 16
PLs     db 0
```



Sets – Add Element

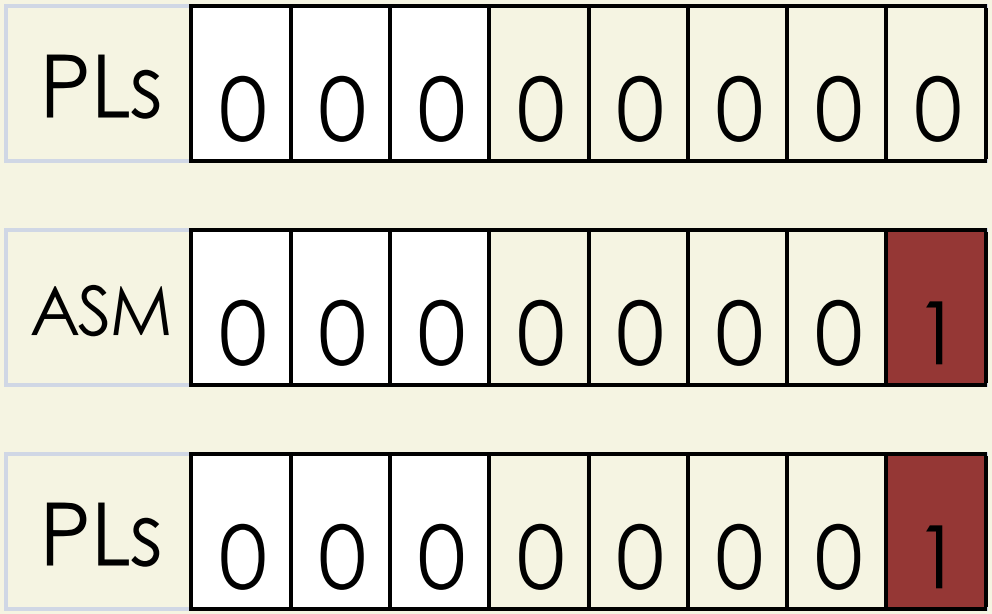
```
ASM     equ 1
C#      equ 2
COBOL   equ 4
JAVA    equ 8
LISP    equ 16
PLs     db 0
```



Sets – Add Element

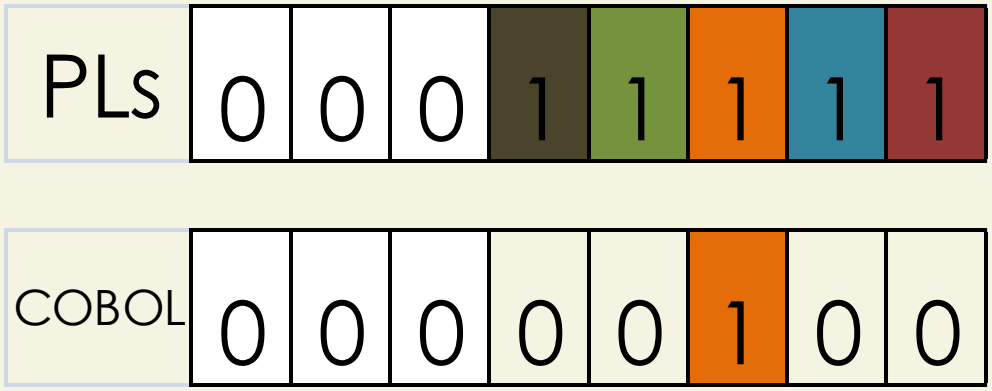
```
ASM     equ 1
C#      equ 2
COBOL   equ 4
JAVA    equ 8
LISP    equ 16
PLs     db 0

OR byte[PLs], ASM
```



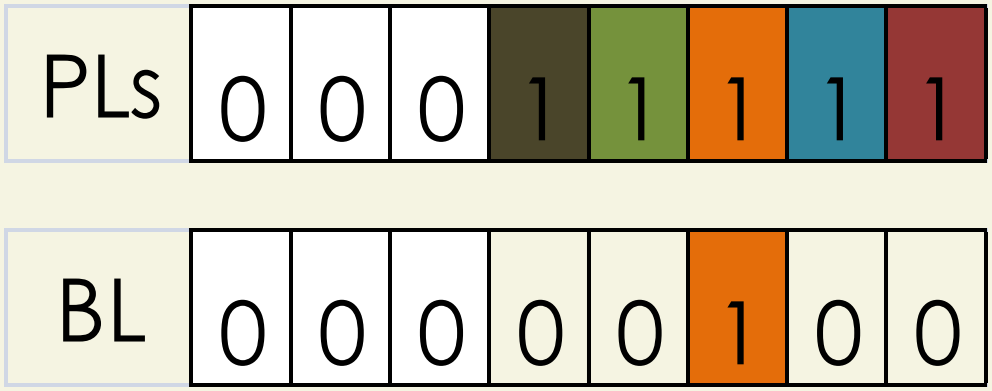
Sets – Remove Element

```
ASM     equ 1
C#      equ 2
COBOL   equ 4
JAVA    equ 8
LISP    equ 16
PLs     db 0
```



Sets – Remove Element

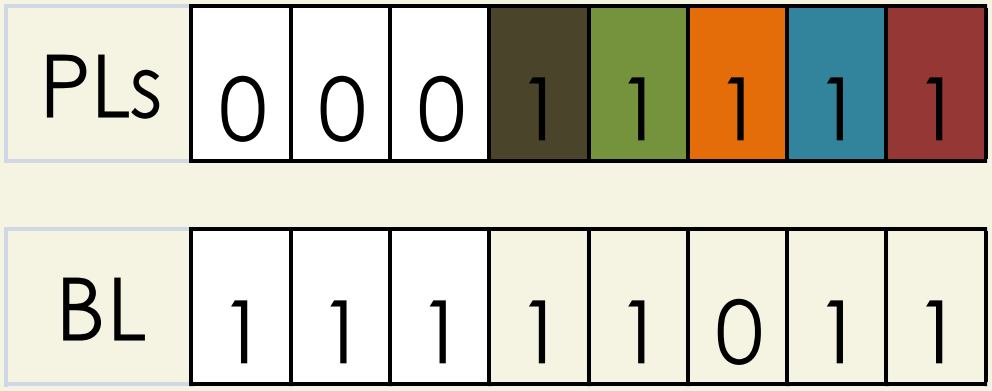
```
ASM     equ 1
C#      equ 2
COBOL   equ 4
JAVA    equ 8
LISP    equ 16
PLs     db 0
```



MOV BL, COBOL

Sets – Remove Element

```
ASM     equ 1
C#      equ 2
COBOL   equ 4
JAVA    equ 8
LISP    equ 16
PLs     db 0
```

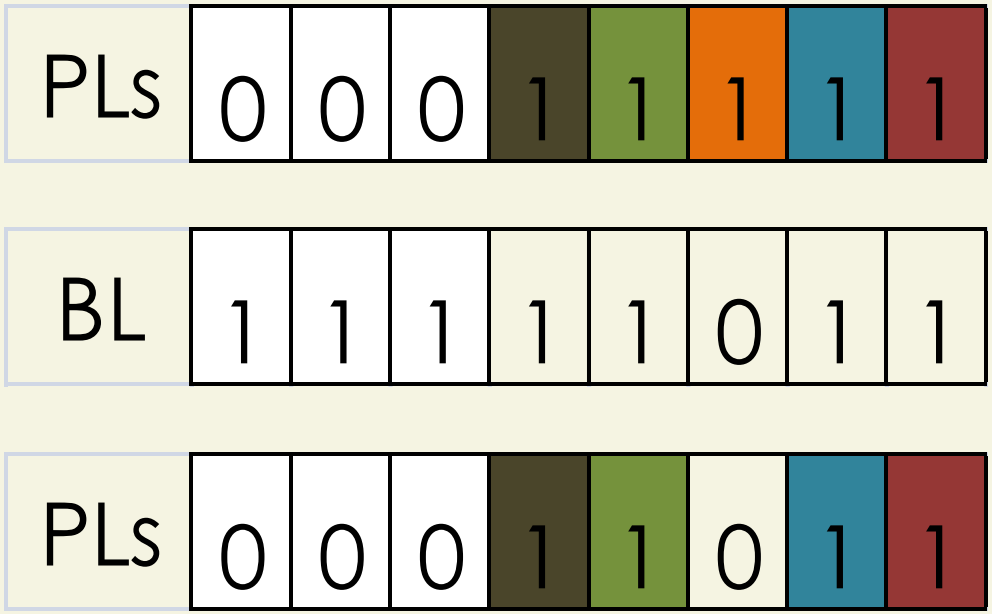


MOV BL, COBOL

NOT BL

Sets – Remove Element

```
ASM     equ 1
C#      equ 2
COBOL   equ 4
JAVA    equ 8
LISP    equ 16
PLs     db 0
```



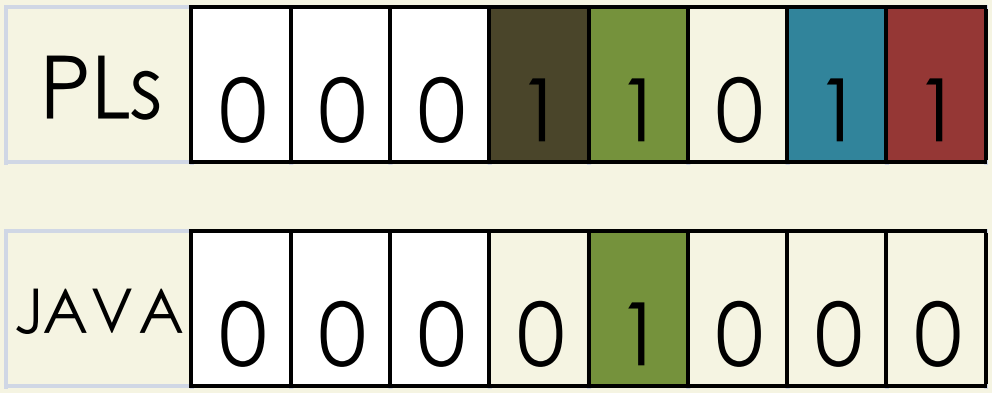
MOV BL, COBOL

NOT BL

AND byte[PLs], BL

Sets – Is Element

```
ASM     equ 1
C#      equ 2
COBOL   equ 4
JAVA    equ 8
LISP    equ 16
PLs     db 0
```



Sets – Is Element

```
ASM     equ 1
C#      equ 2
COBOL   equ 4
JAVA    equ 8
LISP    equ 16
PLs     db 0

TEST byte[PLs], JAVA
JZ noJava
```