

# Code Examples (C and SASM)

```
Int    d1[10];  
D1[3] = 5;
```

## Indirect:

```
lea  addr1, d1  
iadd addr1, 12    ; col-index*element_size =  
                  ; 3*4 = 12  
move m(addr1), 5
```

## Register Indirect:

```
lea  reg1, d1  
iadd reg1, 12    ; col-index*element_size =  
                  ; 3*4 = 12  
move [reg1], 5
```

# Code Examples (C and SASM)

## Based-Displacement:

```
lea reg1, d1  
move 12[reg1], 5
```

## Register Indirect:

```
lea reg1, d1  
move reg2, 12  
move [reg2][reg1], 5
```

# Code Examples (C and SASM)

```
Int    d1[10][6];  
D1[3][4] = 5;
```

## Indirect:

```
lea    addr1, d1
```

```
iadd  addr1, 88
```

```
    ; #columns*row_index*element_size
```

```
    ; + column_index*element_size
```

```
    ; = 6 * 3 * 4 + 4 * 4 = 88
```

```
move  m(addr1), 5
```

# Code Examples (C and SASM)

## Register Indirect:

```
lea reg1, d1  
iadd reg1, 88  
move [reg1], 5
```

## Based-Displacement:

```
lea reg1, d1  
move 88[reg1], 5
```

## Register Indirect:

```
lea reg1, d1  
move reg2, 88  
move [reg2][reg1], 5
```

# Code Examples (C and SASM)

```
Struct {  
    Int    age;  
    Char   gender;  
} s1;  
  
S1.gender = 'f';
```

- Offset = 4
  - size of fields that come before it in the structure
  - each "int" = 4 bytes

# Code Examples (C and SASM)

```
Struct {  
    Int    age;  
    Char   gender;  
} s1[10];  
  
S1[5].gender = 'f';
```

- Offset = 24
  - $\text{column\_index} * \text{element\_size} + \text{size of fields the come before in the structure}$
  - $\text{Column\_index} = 5$
  - $\text{Size of element} = 5 = 4+1$
  - $\text{Fields before "gender"} = 4 \text{ bytes}$
  - $= 5*5 + 4 = 24$