

WORD and SWORD

SWORD AND WORD

```
word1 WORD 65535           ;largest unsigned value
word3 WORD ?               ;uninitialized, unsigned
word2 SWORD -32768         ;smallest signed value

val1 DW 65535              ;Define Word unsigned
val2 DW -32768             ;Signed
```

The **WORD directive** is used to create storage for one or more 16-bit unsigned integers. Each WORD declaration reserves 2 bytes of memory. For example:

- **word1 WORD 65535** reserves 2 bytes for word1 and initializes it with the largest possible unsigned 16-bit value, which is 65535.
- **word3 WORD ?** reserves 2 bytes for word3 but leaves it uninitialized.

SWORD (Define Signed Word): The SWORD directive is similar but is used for signed 16-bit integers. It also reserves 2 bytes of memory. For example:

- **word2 SWORD -32768** reserves 2 bytes for word2 and initializes it with the smallest possible signed 16-bit value, which is -32768.

WORD ARRAYS

Arrays of 16-Bit Words: You can create arrays of 16-bit words by listing the elements or using the DUP operator. For example:

```
myList WORD 1,2,3,4,5
;array to hold 16-bit integers/words
;each element holds 2 bytes(16-bits), so 10 bytes total memory.
;to get the next value, increment by 2.
```

For example, if myList starts at offset 0000, the next element would be at offset 0002, and so on.

This way, you can define and work with arrays of 16-bit words in assembly language, either with initialized values or uninitialized storage.

Offset	Value
0000:	1
0002:	2
0004:	3
0006:	4
0008:	5

Figure above shows a diagram of this array in memory, assuming myList starts at offset 0000. The offsets increment by 4.

Creating arrays with DUP:

```
;Use DUP operator to declare an array with uninitialized values.  
myArrayList2 WORD 5 DUP(?)
```

This code creates an myArrayList2 named array containing 5 uninitialized 16-bit words. Each element of the array occupies 2 bytes of memory.

The ? symbol indicates that the elements are uninitialized, meaning their values are undefined until explicitly set in your code.