**Assembly Language Command**

**32-Bits Register**

**EAX,EBX,ECX.EDX**

**Loop Register = ECX**

**Print data = EAX**

**16-Bits Register**

**AX, BX, CX, DX**

|  |  |
| --- | --- |
| **Data Type** | **Description** |
| Byte/SBYTE | 1 Byte = 8 Bits 256 |
| Word/SWORD | 2 Byte = 16 Bits 65536 |
| DWORD | 4 Byte = 32 Bits >4\*10^9 |
| QWORD | 8 byte = 64 Bits >1\*10^19 |

|  |  |
| --- | --- |
| **Print Methods** | **Description** |
| **Call WriteInt** | **Print signed value** |
| **Call WriteHex** | **Print Hexadecimal** |
| **Call WriteDec** | **Print Unsign value** |

|  |  |
| --- | --- |
| **Command** | **Description** |
| Mov ax , 23h | Move 23 hexadecimal to ax register |
| Mov bx, 03h | Move 03 hexadecimal to bx |
| Add bx, 50 | Move 50 to bx register |
| Sub dx, 70 | Subtract 70 to dx |
| Mul ax, 70 | Multiply 70 to ax |
| imul ax,-20 | Sign Value then need to sign multiply with (+ve,-ve) |
| Div ax, 4 | Divide 4 to ax |
| Idiv ax,-2 | Divide something with sign value with (+ve,-ve) |
| MOV ECX,3 | Loop Three Time |
| L1: | Pointer Looping for L1 |
| LOOP L1 | Return back the Pointer |
| Variable dword 7000,1000 | Declare variable and data type |
| Move esi , offset list | move offset list address into esi register |
| Mov ecx , lengthof Variable | Loop the size of variable e.g 16 |
| Xor eax,eax | if eax equal to eax then become 0  ;Clear Rubbish |
| test dword ptr [esi] , 1 | ; test instruction work similarly to AND operator but do not return value to the destination  ; refer to memory value if without [] become address |
| Jz display | Jump to display function if zero |
| Jc Error | Jump if error |
| Add esi, type Variable | to increase the pointer to the next element --> (add esi,4 Type will declare data type of value) |
| Display: | Display function use display call function |
| Mov eax, dword ptr [eax] | pass the value to eax |
| Call Crlf | go to next line /n |
| xchg eax,[array+4] | Exchange the value |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| **Keyword** | **Description** |
| BYTE, DB (byte) | Allocates unsigned numbers from 0 to 255. |
| SBYTE (signed byte) | Allocates signed numbers from -128 to +127. |
| WORD, DW (word = 2 bytes) | Allocates unsigned numbers from 0 to 65,535 (64K). |
| SWORD (signed word) | Allocates signed numbers from -32,768 to +32,767. |
| DWORD, DD (doubleword = 4 bytes) | Allocates unsigned numbers from 0 to 4,294,967,295 (4 megabytes) |
| SDWORD (signed doubleword) | Allocates signed numbers from -2,147,483,648 to +2,147,483,647. |
| FWORD, DF (farword = 6 bytes) | Allocates 6-byte (48-bit) integers. These values are normally used only as pointer variables on the 80386/486 processors. |
| QWORD, DQ (quadword = 8 bytes) | Allocates 8-byte integers used with 8087-family coprocessor instructions. |
| TBYTE, DT (10 bytes) | Allocates 10-byte (80-bit) integers if the initializer has a radix specifying the base of the number. |