## The `for` statement

- The **`for`** statement is nothing more than a "dressed up" **`while`** statement:

```
for ( expr1; condition; expr2 )     <==>     expr1;
    statements                               while ( condition )
                                             {
                                                statements;
                                                expr2;
                                             }
```

Translate all **for** statements into a **while** statement before translating into assembler code.

- **For-loop example 1**

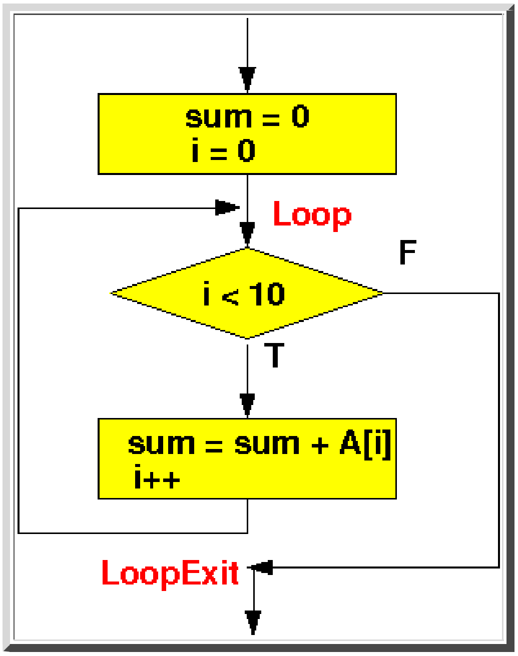
  - **Example:** sum an array

```
int A[10];
int sum, i;

sum = 0;
for (i = 0; i < 10; i++);
    sum = sum + A[i];
```

Convert to **while** loop and then to assembler code:

```
Java:                                  M68000:

sum = 0;                            MOVE.L  #0, sum

i = 0;                              MOVE.L  #0, i
while (i < 10)               Loop:  MOVE.L  i, D0
{                                   CMP.L   #10, D0
   sum = sum + A[i];                BGE     LoopExit
   i++;
}                                   MOVE.L  sum, D0
                                    MOVEA.L  #A, A0
                                    MOVE.L    i, D1
                                    MULS     #4, D1
                                    MOVE.L  0(A0,D1), D2     ; D2 = A[i]
                                    ADD.L   D0, D2           ; D2 = sum + A[i]
                                    MOVE.L  D2, sum          ; Write back to sum in memory

                                    MOVE.L   i, D0
                                    ADD.L   #1, D0
                                    MOVE.L  D0, i

                                    BRA     Loop

                         LoopExit:
```

The flow chart of the **above program** is:



- Here is the assembler program, you can assemble it yourself: click here

  Here is a speedier version: click here

---

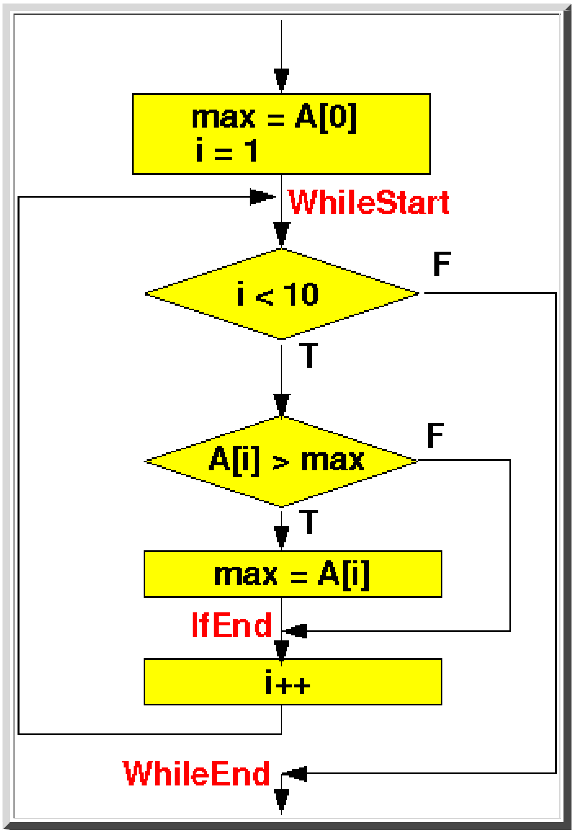
- **For-loop example 2**
  - **Example:** find the **maximum** value in an array

```
int A[10];
int max, i;

max = A[0];
for ( i = 1; i < 10; i++ )
{
   if ( A[i] > max )
      max = A[i];
}
```

Convert to **while** loop:

```
int A[10];
int max, i;

max = A[0];
i = 1;
while ( i < 10 )
{
   if ( A[i] > max )
      max = A[i];

   i++;
}
```

The **flow chart** of the **above program** is:



In **M68000:**

```
      move.l A, max;          * max = A[0] (it's at A !)
      move.l #1, i            * i = 1

WhileStart:
      move.l i,d0
      cmp.l  #10,d0
      bge    WhileEnd

      move.l #A,a0
      move.l i, d0
      muls   #4,d0
      move.l 0(a0,d0), d0     * d0 = A[i]

      move.l max, d1          * d1 = max

      cmp.l  d1, d0           * Compare A[i] ?? max
      ble    IfEnd            * If ( A[i] <= max ), then: bra IfEnd
                                (Because: "not >"    is    "<=" )

      move.l d0, max          * max = A[i]

IfEnd:
      move.l i, d0
      add.l  #1, d0
      move.l d0, i

      bra    WhileStart
WhileEnd:
```

- Here is the assembler program, you can assemble it yourself: click here

  Here is the **Egtapi** debug file for the program: click here