Execution Times

|  |  |  |
| --- | --- | --- |
| **Function** | **Assembly [µs]** | **C [µs]** |
| Generate unique number | 2 | 7 |
| Generate checksum | 6 | 13.5 |
| Validate checksum | 4.25 | 7.5 |
| Bubble sort | 24 | 113 |
| Toggle LED Assembly - 0.65 (this should technically be subtracted from all results) | | |

**int** **calculate\_unique\_number\_in\_c**(**int** a, **int** b, **int** c, **int** d, **int** e, **int** f, **int** g, **int** h)

{

//**TODO**: write your code here

**int** smallTot = 0, bigTot = 0, Tot = 0;

**char** num[] = {a, b, c, d, e, f, g, h};

bubble\_sort\_in\_c(num);

smallTot = num[0] + num[1] + num[2] + num[3];

bigTot = num[4] + num[5] + num[6] + num[7];

Tot = bigTot \* smallTot;

**return** Tot;

}

**int** **generate\_checksum\_in\_c**(**char**\* studentNumber)

{

//**TODO**: write your code here

**int** i;

**int** N = 8;

**int** tot = 0;

**int** tempNum = 0;

**for**(i = 0; i < N; i++)

{

tempNum = studentNumber[i] - 48;

tot += tempNum \* (8 - i);

}

**return** tot;

}

**int** **validate\_checksum\_in\_c**(**int** checksum)

{

//**TODO**: write your code here

**return** (checksum % 11)? 0 : 1;

}

**void** **bubble\_sort\_in\_c**(**char**\* studentNumber)

{

//**TODO**: write your code here

**int** i;

**int** j;

**int** N = 8;

**for**(i = 0; i < N; i++)

{

**for**(j = 0; j < N; j++)

{

**if**(studentNumber[i] > studentNumber[j])

swap(i, j, studentNumber);

}

}

}

**void** **swap**(**int** first, **int** second, **char**\* array)

{

**char** temp = array[first];

array[first] = array[second];

array[second] = temp;

}

The times vary fairly drastically between the C and assembly code as the compiler that translates the C into assembly writes far more lines than is necessary for the actual assembly code. Thus as I wrote the assembly myself and know that is as essentially as close to the most efficient that it can be, it is much faster than the C code.

I would rather code in C for the convenience and speed at which you can write the few lines of code.

I would rather code in Assembly when I need to do many calculations and simulations as it is much more efficient than C (or other high level programing languages for that matter). Even though it takes long to write the code and it is many lines and difficult to read interpret and debug, it is much more efficient.