

[bhi, lea, math](#)

```
== apollo_asm ==
```

```
* this program calculate prime numbers and store them in an array
* credits to pisklak for the code
max_number EQU 20

        lea    fib_array,a0        ; load fib_array addr into a0
        move.l #2,d1               ; setup counter reg
fill_loop
        move.l d1,(a0)+            ; write value to array
        add.l  #1,d1               ; increasing counter
        cmp.l  #max_number,d1      ; checking do we are not at the end
of array
        bls   fill_loop           ; if not then execute fill loop again

        lea    fib_array,a0        ; reset array addr in a0
        move.l #2,d2               ; starting position in d2
main_loop
        cmp.l  #max_number,d2      ; check do we not at last position
        bhi   endprg
        move.l -8(a0,d2.l*4),d0     ; we get our number from array
        cmp.l  #0,d0               ; and we check against our "throw"
sign
        beq   check_next          ; if it is thrown one then move on to next
        move.l d0,d1               ; clone it in d1
throw_loop
        add.l  d0,d1               ; now we have in d1 next multiply of our
number
        cmp.l  #max_number,d1      ; we check do we do not exceed our
range
        bhi   check_next          ; if yes then we throw out all multiplies
and should
                                ; check next number from array
        move.l #0,d0               ; mark 'thrown' sign
        bra   throw_loop          ; throw next multiply
check_next
        add.l  #1,d2               ; move to next number
        bra   main_loop
endprg RTS

fib_array DS.l max_number
```

From:

<https://wiki.apollo-accelerators.com/> - **Apollo Accelerators Public Wiki**

Permanent link:

[https://wiki.apollo-accelerators.com/doku.php/prime\\_number\\_calculator](https://wiki.apollo-accelerators.com/doku.php/prime_number_calculator)

Last update: **2016/08/31 11:32**

