

**DIGITAL ASSIGNMENT : general instructions**

**MICROPROCESSOR AND INTERFACING(D1)[MRS SHOBHA REKH]**



**September 13, 2022**

**ANIRUDH VADERA**

**20BCE2940**

**REGISTRATION NUMBER: 20BCE2940**

**QUESTION TO BE CHOSEN: 2 + 0 + 2 + 9 + 4 + 0 = 17 = 1 + 7 = 8**

**QUESTION 8 Part A:**

**Find if the given number is a prime number.**

**Algorithm:**

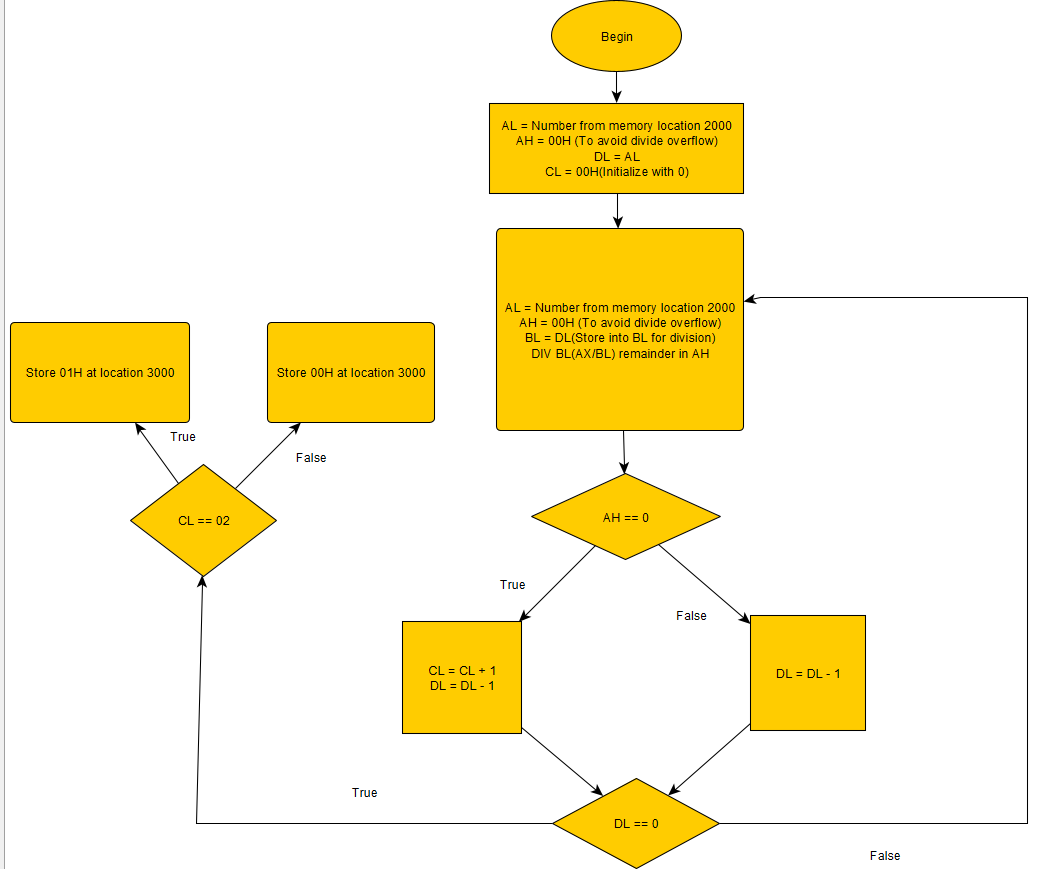
1. Take n as input
2. Run a loop from i = n to 1. For each iteration, check if i divides n completely or not. If it does, then i is n’s divisor
3. Keep a count of the total number of divisors of n
4. If the count of divisors is 2, then the number is prime, else composite

**In Assembly Language:**

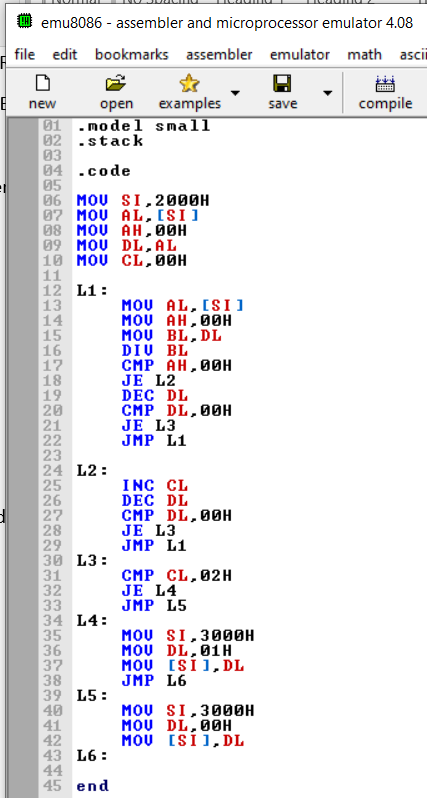
DL goes from n to 1 each time we divide the given number by DL(Actually the given number is in AL then AH is 00H(to avoid divide overflow) and we move DL to BL then execute DIV BL)

1. Load the data from memory from location 2000 into AL register
2. Move 00H into AH in order to avoid Divide overflow error
3. Move AL value into DL register which will be decremented later
4. Move 00H into CL register it stores the number of divisors of the given number at the end if the value of CL is 02H then the given number is prime else its not a prime number
5. We start with label L1:
6. We take the number into AL and move 00H into AH
7. Then move DL into BL in order to perform division operation
8. Compare the value of AH(AH Stores the remainder of division) with 00H after division
9. Use JE(Jump on equal) after CMP to label2 now if the number has a divisor then the value of CL will be incremented
10. In L2 we just increment the value of CL
11. We then decrease DL by 1
12. If DL is 00H then we jump to L3 otherwise we loop back again in L1
13. In L3 we simply compare CL by 02H if it is 02H then we jump to L4 otherwise to L5
14. In L4 it means the number of divisors of the given number is only 2 hence a number is prime hence we store the value 01H at location 3000H
15. In L5 it means the number of divisors of the given number is more than 2 hence a number is not prime hence we store the value 00H at location 3000H
16. We end the program now

**FlowChart:**



**Code Screenshot:**



**Assembly Language Code:**

.model small

.stack

.code

MOV SI,2000H

MOV AL,[SI]

MOV AH,00H

MOV DL,AL

MOV CL,00H

L1:

MOV AL,[SI]

MOV AH,00H

MOV BL,DL

DIV BL

CMP AH,00H

JE L2

DEC DL

CMP DL,00H

JE L3

JMP L1

L2:

INC CL

DEC DL

CMP DL,00H

JE L3

JMP L1

L3:

CMP CL,02H

JE L4

JMP L5

L4:

MOV SI,3000H

MOV DL,01H

MOV [SI],DL

JMP L6

L5:

MOV SI,3000H

MOV DL,00H

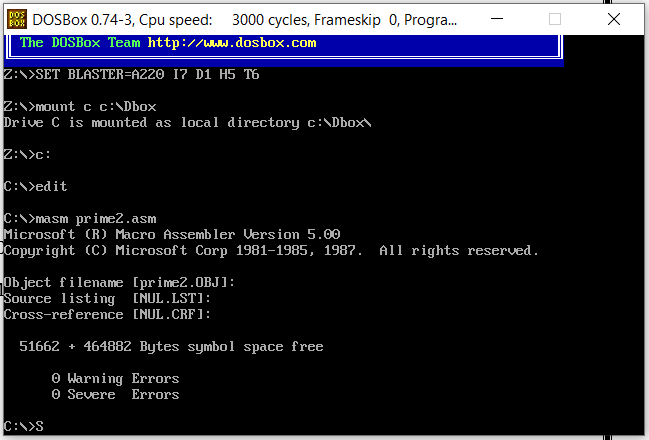
MOV [SI],DL

L6:

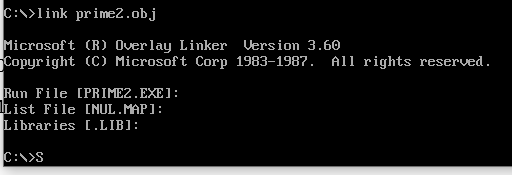
end

**Execution Proof:**

**Command: masm prime2.asm**



**Command: link prime2.obj**



**Command: debug prime2.exe**



**Given Input at location 2000H**

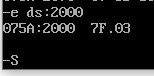
**Output at Location 3000H**

**If value stored at location 3000H is 00H – The Given Number is not prime**

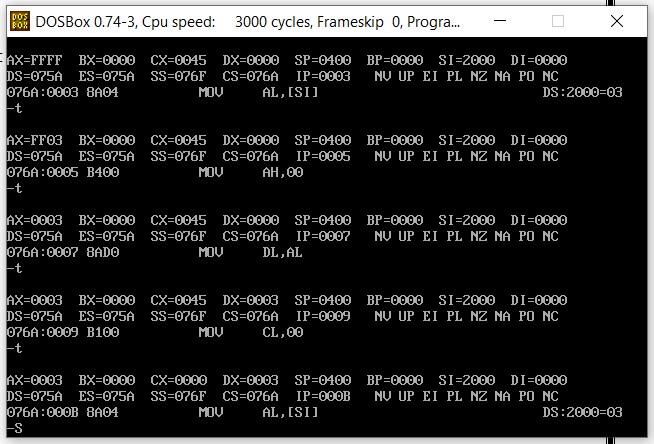
**If value stored at location 3000H is 01H – The Given Number is prime**

**Giving Inputs:**

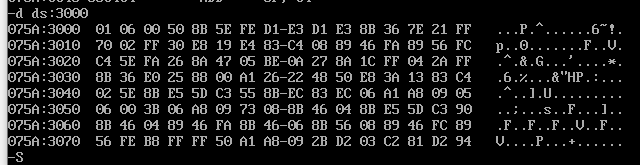
Given Number : **03** (Expected Output **01H** at 3000H)



**A screenshot of process in middle:**



**The Output is:**

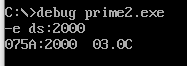




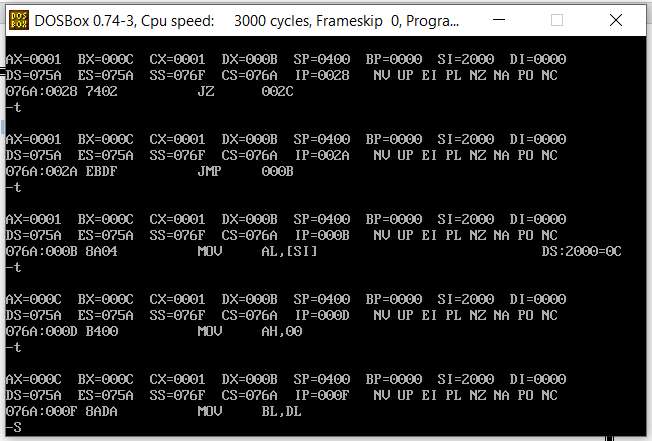
Hence **01H** is stored at **location 3000** hence **03 is a prime number**

**Giving Inputs:**

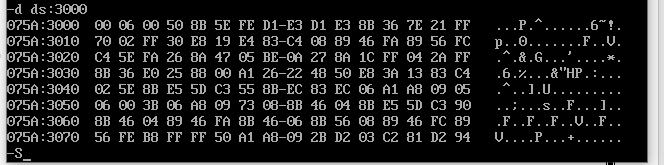
Given Number : **12(0C)** (Expected Output **00H at 3000H**)



**A screenshot of process in middle:**



**The Output is:**





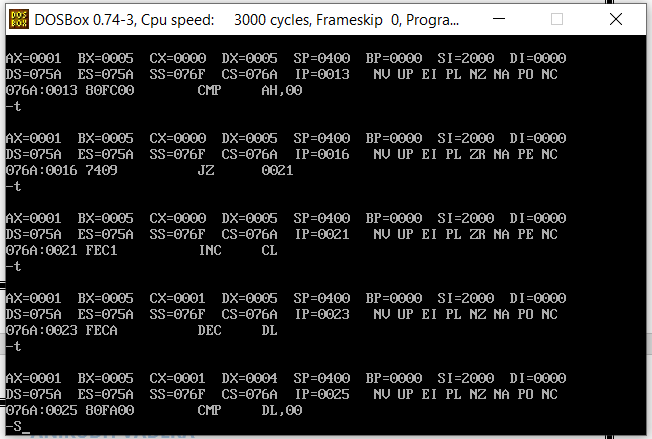
Hence **00H** is stored at **location 3000** hence **12(0C) is not a prime number**

**Giving Inputs:**

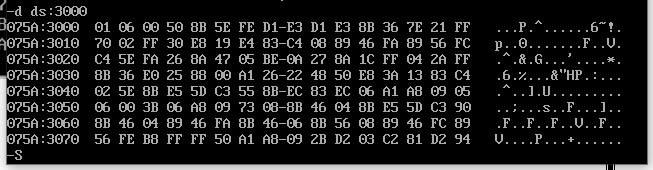
Given Number : **05** (Expected **Output 01H at 3000H**)



**A screenshot of process in middle:**



**The Output is:**





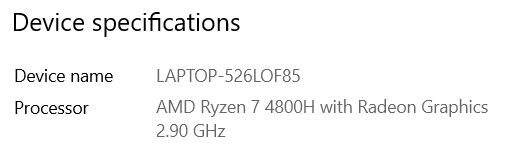
Hence **01H** is stored at **location 3000** hence **05 is a prime number**

**QUESTION 8 Part B:**

**Find the name of the processor available in your laptop. Identify the size of address and data.**

**Solution:**

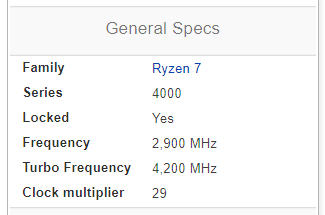
**Name of the Processor: AMD Renoir (Ryzen 4800 APU)**

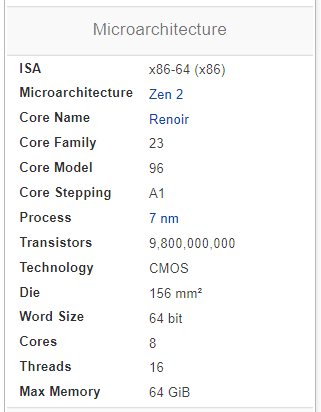


**CodeName(Micro Architecture Used): Renoir (Zen 2)**

**Short Description:**

**Ryzen 7 4800H** is a 64-bit octa-core high-end performance x86 mobile microprocessor introduced by AMD in early 2020. Fabricated on TSMC's 7-nanometer process and based on AMD's Zen 2 microarchitecture, the 4800H operates at a base frequency of 2.9 GHz with a TDP of 45 W and a boost frequency of up to 4.2 GHz.





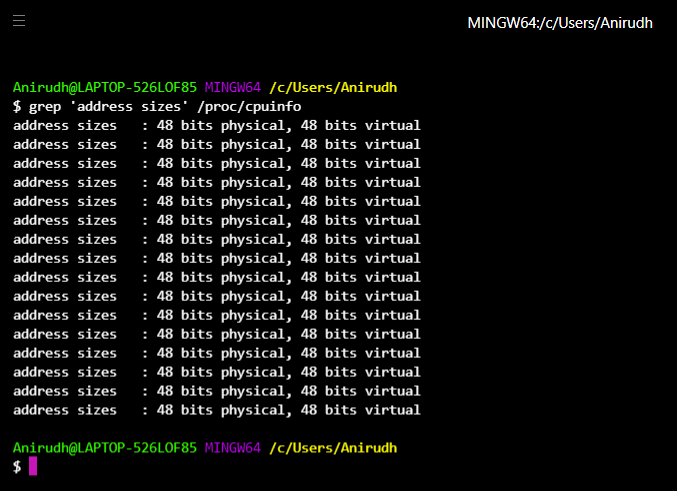
**Address and Data Bus(Size Space) :**

**Given ISA : x86-64 (x86)**

There are not yet any x86-64 systems that have a 64 bit address bus.

In order to find address bus size type use the following command:

**grep ‘address sizes’ /proc/cpuinfo**

****

**We get to know that the address bus is of 48 bits**

* **At most 248 of bytes can be addressed for ROM, RAM, VRAM, IO ….**
* **At most 248 of bytes of virtual memory can be addressed, per process.**

**Space = 248B = 28 TB = 256TB**

**Data Bus(Space):**

**Given ISA : x86-64 (x86)**

The logical width of the data-bus, of an x86-64 is simply **64 bits(8-byte wide data bus).**

However, the physical size is whatever the manufacturer chooses. They can choose whatever they want, without affecting behaviour Eg. Multiplex 64 bits over 32 or 16 bits, or send two 64 bits over 32 or 16 bits, or send two 64 bit values over a 128-bit bus.

Maximum Possible (Data)Memory Spaces available: 2^64 = **1.8446744e+19 (18.4 exabytes) But in reality its way less than that.**

**Space = 264B = 24 Exabyte = 16EB**