```
Austins-MacBook-Pro:proj1 excalibur$ make d
gcc disassembler.c -Wall -g -o diss && ./diss

<--- Machine instruction (in hex): 0x22da822 --->

Memory Address: 0x7a060
Opcode hex value: 0x0
Instruction type: R Format
First register (source) hex value: 0x11
Second register (source) hex value: 0xd
Third register (destination) hex value: 0x15
Function value: 0x22
Function name: sub

---> Assembly instruction: 0x7a060: sub $21, $17, $13 <---


<--- Machine instruction (in hex): 0x8ef30018 --->

Memory Address: 0x7a064
Opcode hex value: 0x23
Instruction type: I Format
First register (source) hex value: 0x17
Second register (source/dest) hex value: 0x13
Constant/Offset hex value: 0x18
Operation name: lw

---> Assembly instruction: 0x7a064: lw $19,24($23) <---


<--- Machine instruction (in hex): 0x12a70004 --->

Memory Address: 0x7a068
Opcode hex value: 0x4
Instruction type: I Format
First register (source) hex value: 0x15
Second register (source/dest) hex value: 0x7
Constant/Offset hex value: 0x4
Operation name: beq

---> Assembly instruction: 0x7a068: beq $21,$7 (branch to: 0x7a07c) <---


<--- Machine instruction (in hex): 0x2689820 --->

Memory Address: 0x7a06c
Opcode hex value: 0x0
Instruction type: R Format
First register (source) hex value: 0x13
```

```
Second register (source) hex value: 0x8
Third register (destination) hex value: 0x13
Function value: 0x20
Function name: add

---> Assembly instruction: 0x7a06c: add $19, $19, $8 <---


<--- Machine instruction (in hex): 0xad930018 --->

Memory Address: 0x7a070
Opcode hex value: 0x2b
Instruction type: I Format
First register (source) hex value: 0xc
Second register (source/dest) hex value: 0x13
Constant/Offset hex value: 0x18
Operation name: sw

---> Assembly instruction: 0x7a070: sw $19,24($12) <---


<--- Machine instruction (in hex): 0x2697824 --->

Memory Address: 0x7a074
Opcode hex value: 0x0
Instruction type: R Format
First register (source) hex value: 0x13
Second register (source) hex value: 0x9
Third register (destination) hex value: 0xf
Function value: 0x24
Function name: and

---> Assembly instruction: 0x7a074: and $15, $19, $9 <---


<--- Machine instruction (in hex): 0xad8ffff4 --->

Memory Address: 0x7a078
Opcode hex value: 0x2b
Instruction type: I Format
First register (source) hex value: 0xc
Second register (source/dest) hex value: 0xf
Constant/Offset hex value: 0xfffffff4
Operation name: sw

---> Assembly instruction: 0x7a078: sw $15,-12($12) <---


<--- Machine instruction (in hex): 0x18c6020 --->
```

Memory Address: 0x7a07c
Opcode hex value: 0x0
Instruction type: R Format
First register (source) hex value: 0xc
Second register (source) hex value: 0xc
Third register (destination) hex value: 0xc
Function value: 0x20
Function name: add

---> Assembly instruction: 0x7a07c: add $12, $12, $12 <---

<--- Machine instruction (in hex): 0x2a4a825 --->

Memory Address: 0x7a080
Opcode hex value: 0x0
Instruction type: R Format
First register (source) hex value: 0x15
Second register (source) hex value: 0x4
Third register (destination) hex value: 0x15
Function value: 0x25
Function name: or

---> Assembly instruction: 0x7a080: or $21, $21, $4 <---

<--- Machine instruction (in hex): 0x158ffff6 --->

Memory Address: 0x7a084
Opcode hex value: 0x5
Instruction type: I Format
First register (source) hex value: 0xc
Second register (source/dest) hex value: 0xf
Constant/Offset hex value: 0xfffffff6
Operation name: bne

---> Assembly instruction: 0x7a084: bne $12,$15 (branch to: 0x7a060) <---

<--- Machine instruction (in hex): 0x8e59fff0 --->

Memory Address: 0x7a088
Opcode hex value: 0x23
Instruction type: I Format
First register (source) hex value: 0x12
Second register (source/dest) hex value: 0x19
Constant/Offset hex value: 0xfffffff0
Operation name: lw

```
---> Assembly instruction: 0x7a088: lw $25,-16($18) <---

Austins-MacBook-Pro:proj1 excalibur$
```