Group 21 COMP4300: Project 1 – Encoding Jared Brown
Adam Eichelkraut

1. Stack Instruction Set:

Instructions	Opcode (8-bits)	Address (24-bits)
PUSH	0x01	Source
POP	0x02	Destination
MULT	0x03	N/A
ADD	0x04	N/A
END	0x05	N/A

Accumulator Instruction Set:

Instructions	Opcode (8-bits)	Address (24-bits)
LOAD	0x01	Source
STO	0x02	Destination
MULT	0x03	Source
ADD	0x04	Source
END	0x05	N/A

Segment Address Offsets:
User Text = 0x00100000 – 0x00101000
User Data = 0x00200000 – 0x00201000
Kernel Text = 0x00300000 – 0x00301000
Kernel Data = 0x00400000 – 0x00401000
Stack = 0x00500000 – 0x00501000

2. Stack Source Code:

```
#Source Code
.data
0x00200000:3
0x00200004:7
0x00200008:5
0x0020000C:4

.text
 PUSH 0x00200000
 PUSH 0x00200000
 MULT
 PUSH 0x00200004
 MULT
 PUSH 0x00200000
 PUSH 0x00200008
 MULT
 ADD
 PUSH 0x0020000C
 ADD
 POP 0x00200010
 END
```

```
#Binary Translation
.data
0x00200000:3
0x00200004:7
0x00200008:5
0x0020000C:4

.text
 0x01200000
 0x01200000
 0x03000000
 0x01200004
 0x03000000
 0x01200000
 0x01200008
 0x03000000
 0x04000000
 0x0120000C
 0x04000000
 0x02200010
 0x05000000
```

Accumulator Source Code:

```
#Source Code
.data
0x00200000:3  #X
0x00200004:7  #A
0x00200008:5  #B
0x0020000C:4  #C

.text
  LOAD 0x00200000
  MULT 0x00200000
  MULT 0x00200004
  STO 0x00200010
  LOAD 0x00200000
  MULT 0x00200008
  ADD 0x00200010
  ADD 0x0020000C
  STO 0x00200010
  END
```

```
#Binary Translation
.data
0x00200000:3  #X
0x00200004:7  #A
0x00200008:5  #B
0x0020000C:4  #C

.text
  0x01200000
  0x03200000
  0x03200004
  0x02200010
  0x01200000
  0x03200008
  0x04200010
  0x0420000C
  0x02200010
  0x05000000
```

3. Size of Quadratic Evaluator Code:

   Stack: 4 lines in .data + 13 in .text = 17 lines of instr * 4-bytes (32-bit instr) = 68 bytes

   Accumulator: 4 lines in .data + 10 in .text = 14 lines of instr * 4-bytes (32-bit instr) = 56 bytes

   MIPS:
   (total) 6 lines in .data + 19 in .text = 25 lines of instr * 4-bytes (32-bit instr) = 100 bytes
   (without print/syscalls) 4 in .data + 9 in .text = 13 lines of instr * 4-bytes (32-bit instr) = 52 bytes