First:_____ Last:_____

This is a closed book exam. You must put your answers on pages 1,2,3,4 only. You have 50 minutes, so allocate your time accordingly. Show your work, and put your answers in the boxes. ***Please read the entire quiz before starting.***

(4) Question 1. logic currently uses binary because it is fast, low power, and very small. In the future, an EE319K student invents ternary logic that is faster, smaller and lower power than binary. This means each ternary bit can be 0, 1, or 2. Ternary means base 3 in the same way binary means base 2. What is the value of the unsigned four-digit ternary number 1021? Give your answer as a decimal number. ----------

(3) Question 2. Answer true/false for each of the following three statements

Part a) The stack pointer (SP) points to the data on top of the stack. ----------
(Assume there is data on the stack. I.e., the stack is not empty.)

Part b) I add three numbers by executing **`adda`** twice. The order in which I add the numbers affects the final value of RegA. ----------

Part c) Dropout error can occur on a logical left shift (e.g., **`lsla`**). ----------

(4) Question 3. Consider the following two instructions

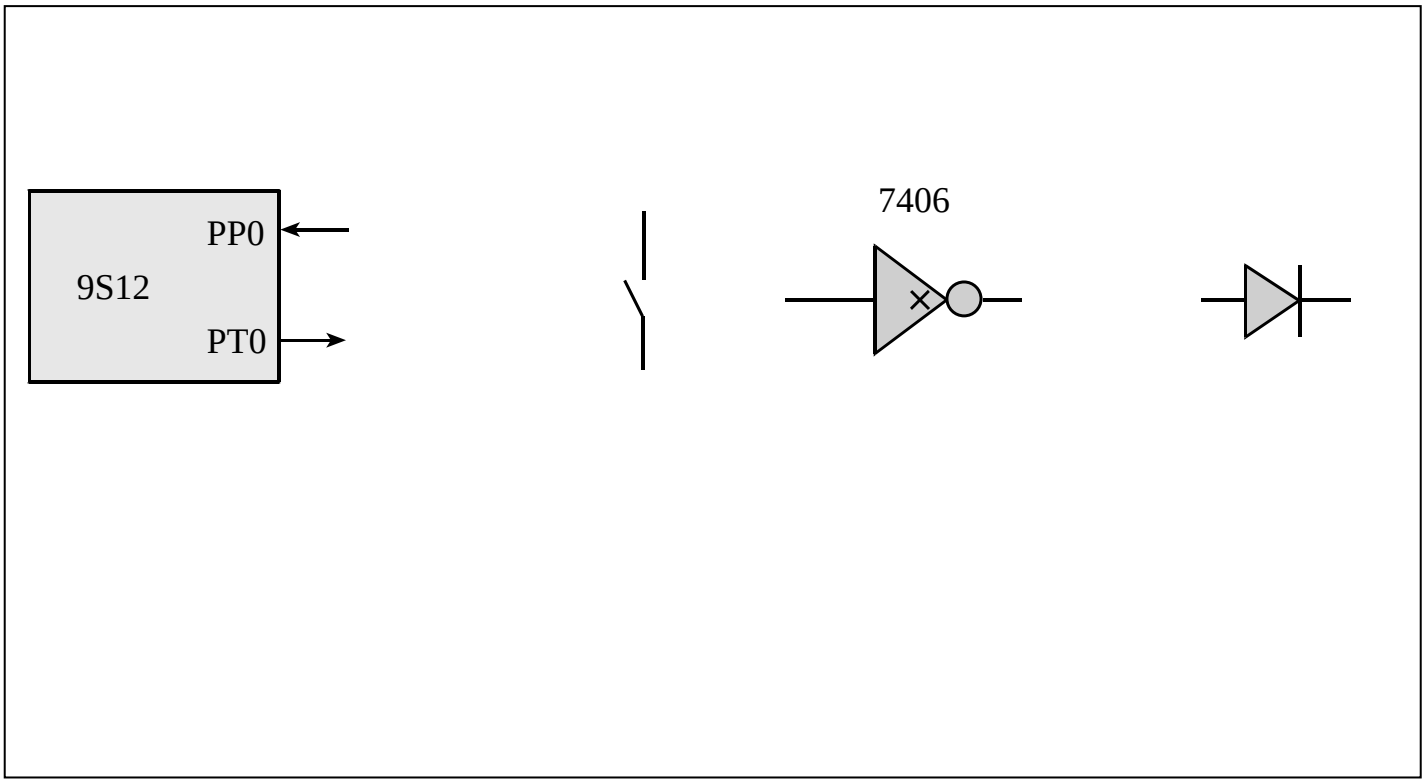
```
ldab #-6
subb #251
```

What will be the value of the overflow (V) bit?

What will be the value of the carry (C) bit?

(4) Question 4. What is the binary representation of 8-bit signed number -11?

(20) Question 5. Interface the LED to PT0. The desired LED operating point is 2.0V at 25 mA. At 25 mA you can assume the V_{OL} of the 7406 will be 0.5 V. Interface the switch to PP0 using positive logic. No software is required in this question, and you may assume PT0 is an output and PP0 an input. Your bag of parts includes the switch, the 7406, the LED, and one resistor each of the values {1Ω, 10Ω, 100Ω, 1kΩ, 10kΩ, 100kΩ and 1MΩ}. Pick the best resistors to use (you will not need them all.)



(5) Question 6. Assume PC is $4200, and Register X is $1234. You may assume all RAM locations are initially 0. Show the simplified bus cycles occurring when the **`stx`** instruction is executed. In the "**changes**" column, specify which registers get modified during that cycle, and the corresponding new values. Do not worry about changes to the CCR. *Just show the one instruction.*

```
$4200 7E3000    stx $3000
```

R/W	Addr	Data	Changes to D,X,Y,S,PC,IR,EAR

For questions 7 8, and 9, don't worry about establishing the reset vector, creating a main program, or initializing the stack pointer. You may use RAM-based global variables. Include comments. You may use the following definitions

```
PTT   equ  $0240
DDRT  equ  $0242
```

(20) Question 7. Assume seven positive logic switches are connected to PT6-PT0, and one LED is connected to PT7. Assume the direction register is properly initialized. Write an assembly language subroutine that sets PT7=1, if PT0=1, PT2=0, and PT6=0, regardless of the other 4 switches. For all other patterns of input switches, do not change the PT7 output.

(20) Question 8. Write an assembly language subroutine that adds two unsigned 16-bit numbers. The two inputs are passed in Register X and Register Y, and the result is returned in Register D. Implement ceiling, such that if the sum is too big for 16 bits, return 65535.

(20) Question 9. Write an assembly language subroutine that counts the number of binary bits that are zero in an 8-bit number. The 8-bit input parameter is passed in Register A and the result is returned in Register B. For example, if Register A = %00000001, return Register B=7 because there are 7 binary zeros. If Register A = %11111001, return Register B =2 because there are 2 binary zeros.

STX Store Index Register X STX

Operation: $(X_H : X_L) \Rightarrow M : M + 1$

Description: Stores the content of index register X in memory. The most significant byte of X is stored at the specified address, and the least significant byte of X is stored at the next higher byte address (the specified address plus one).

Source Form	Address Mode	Object Code	HCS12 Access Detail
STX opr8a	DIR	**5E dd**	**PW**
STX opr16a	EXT	**7E hh ll**	**PWO**
STX oprx0_xysp	IDX	**6E xb**	**PW**
STX oprx9,xyssp	IDX1	**6E xb ff**	**PWO**
STX oprx16,xysp	IDX2	**6E xb ee ff**	**PWP**

BSR Branch to Subroutine BSR

Operation: $(SP) - \$0002 \Rightarrow SP$
$RTN_H : RTN_L \Rightarrow M_{(SP)} : M_{(SP+1)}$
$(PC) + Rel \Rightarrow PC$

Description: Sets up conditions to return to normal program flow, then transfers control to a subroutine. Uses the address of the instruction after the BSR as a return address. Decrements the SP by two, to allow the two bytes of the return address to be stacked. Stacks the return address (the SP points to the high-order byte of the return address). Branches to a location determined by the branch offset. Subroutines are normally terminated with an RTS instruction, which restores the return address from the stack.

Source Form	Address Mode	Object Code	Access Detail HCS12
BSR rel8	REL	**07 rr**	**SPPP**

RTS Return from Subroutine RTS

Operation: $(M_{(SP)} : M_{(SP+1)}) \Rightarrow PC_H : PC_L$; $(SP) + \$0002 \Rightarrow SP$

Description: Restores context at the end of a subroutine. Loads the program counter with a 16-bit value pulled from the stack and increments the stack pointer by two. Program execution continues at the address restored from the stack.

Source Form	Address Mode	Object Code	Access Detail HCS12
RTS	INH	**3D**	**UfPPP**

SUBB Subtract B SUBB

Operation: (B) – (M) ⇒ B

Description: Subtracts the content of memory location M from the content of accumulator B and places the result in B. For subtraction instructions, the C status bit represents a borrow.

N: Set if MSB of result is set; cleared otherwise

Z: Set if result is $00; cleared otherwise

V: $B7 \bullet \overline{M7} \bullet \overline{R7} + \overline{B7} \bullet M7 \bullet R7$
Set if a two's complement overflow resulted from the operation; cleared otherwise

C: $\overline{B7} \bullet M7 + M7 \bullet R7 + R7 \bullet \overline{B7}$
Set if the value of the content of memory is larger than the value of the accumulator; cleared otherwise

Source Form	Address Mode	Object Code	Access Detail HCS12
SUBB #*opr8i*	IMM	C0 ii	P
SUBB *opr8a*	DIR	D0 dd	rPf
SUBB *opr16a*	EXT	F0 hh ll	rPO
SUBB *oprx0_xysp*	IDX	E0 xb	rPf
SUBB *oprx9,xysp*	IDX1	E0 xb ff	rPO
SUBB *oprx16,xysp*	IDX2	E0 xb ee ff	frPP
SUBB [D,*xysp*]	[D,IDX]	E0 xb	fIfrPf
SUBB [*oprx16,xysp*]	[IDX2]	E0 xb ee ff	fIPrPf

ADDA Add without Carry to A ADDA

Operation: (A) + (M) ⇒ A

Description: Adds the content of memory location M to accumulator A and places the result in A.

N: Set if MSB of result is set; cleared otherwise

Z: Set if result is $00; cleared otherwise

V: $A7 \bullet M7 \bullet \overline{R7} + \overline{A7} \bullet \overline{M7} \bullet R7$
Set if two's complement overflow resulted from the operation; cleared otherwise

C: $A7 \bullet M7 + M7 \bullet \overline{R7} + \overline{R7} \bullet A7$
Set if there was a carry from the MSB of the result; cleared otherwise