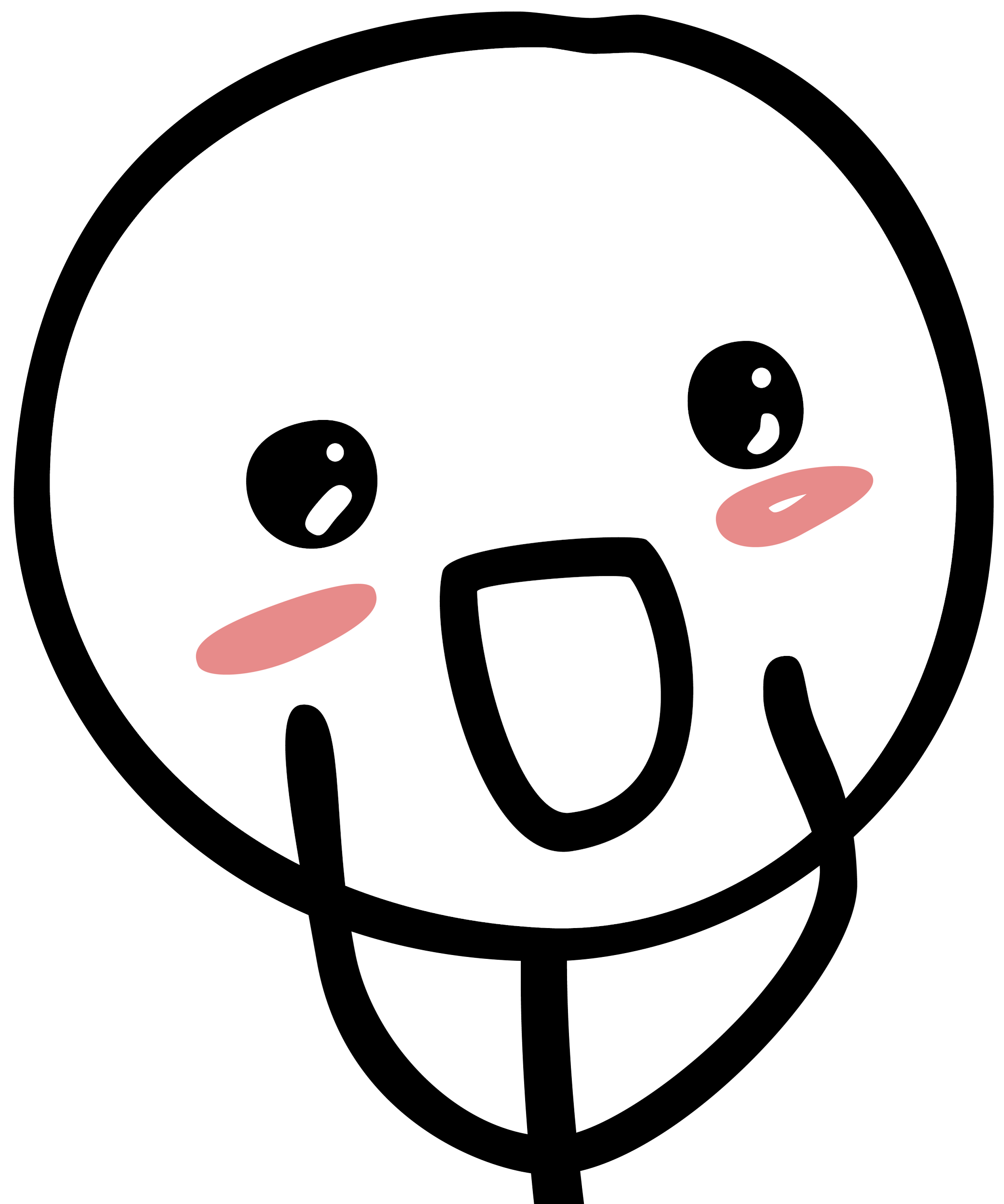
****

**Pre-Lab 6: Teaching the Bear to Navigate the Maze**

Name: Socheath Sok,

Class: EE 346 L

Section 6, T/Th

Date: November 26, 2018

**Questions**

1. What is stack underflow and overflow?

Stack underflow is error that occur when pop is used on an empty stack.

Stack Overflow is is error that occur when more items are being push onto the stack

than it can hold.

1. Which of the subroutine rules, if not followed, would result in a stack underflow and/or overflow condition. Explain your answers.

Registers that are needed for modification must be specified in the code to avoid popping registers from an empty stack and cause underflow.

If registers are not push in the beginning and pop at the end in reverse order, more than desired items might be placed on the stack and cause overflow.

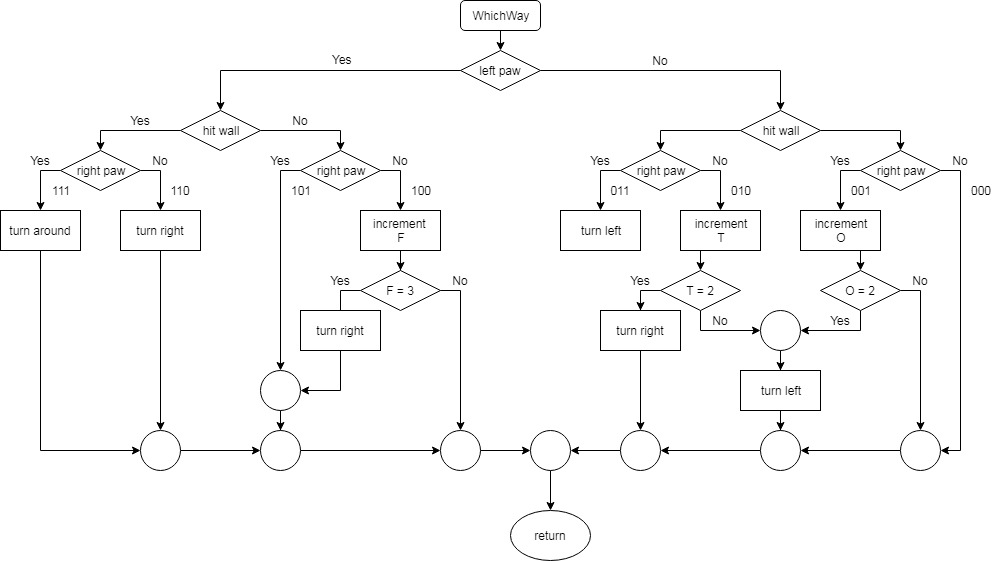
1. How would you represent minus 1 in hexadecimal using 2’s complement notation?

0xFF

1. Write an assembly program to implement the WhichWay flowchart.

| **WhichWay:**  **push r25**  **push r26**  **push r27**  **lds r24,dir**  **rcall LeftPaw**  **tst r24**  **breq case0xx**  **case1xx:**  **lds r24,dir**  **rcall HitWall**  **tst r24**  **breq case10x**  **case11x:**  **lds r24,dir**  **rcall RightPaw**  **tst r24**  **breq case110**  **case111:**  **rjmp TA**  **case110:**  **rjmp TR**  **case10x:**  **lds r24,dir**  **rcall RightPaw**  **tst r24**  **breq case100**  **case101:**  **lds r24,dir**  **rjmp WhichEnd**  **case100:**  **lds r25,N**  **inc r25**  **sts N,r25**  **cpi r25, 0x01**  **brne TR**  **cpi r25, 0x02**  **brne TR**  **cpi r25, 0x04**  **brne TR**  **cpi r25, 0x05**  **brne TR**  **lds r24,dir**  **rjmp whichEnd**  **case0xx:**  **lds r24,dir**  **rcall HitWall**  **tst r24**  **breq case00x** | **case01x:**  **lds r24,dir**  **rcall RightPaw**  **tst r24**  **breq case010**  **case011:**  **rjmp TL**  **case010:**  **lds r26,N**  **inc r26**  **sts N,r26**  **cpi r26,0x01**  **brne TR**  **cpi r26,0x05**  **brne TR**  **cpi r26,0x06**  **brne TR**  **cpi r26,0x07**  **brne TR**  **cpi r26,0x08**  **brne TR**  **rjmp TL**  **case00x:**  **lds r24,dir**  **rcall RightPaw**  **tst r24**  **breq case000**  **case001:**  **lds r27,N**  **inc r27**  **sts N,r27**  **cpi r23,0x01**  **brne TL**  **cpi r23,0x03**  **brne TL**  **lds r24,dir**  **rjmp WhichEnd**  **case000:**  **lds r24,dir**  **rjmp whichEnd**  **TL: lds r24,dir**  **rcall TurnLeft**  **rjmp WhichEnd**  **TR: lds r24,dir**  **rcall TurnRight**  **rjmp WhichEnd**  **TA :lds r24,dir**  **rcall TurnAround**  **rjmp WhichEnd**  **WhichEnd:**  **pop r27**  **pop r26**  **pop r25**  **ret** |
| --- | --- |

**Flowchart**

****