**Assignment # 2 (submission due date 23/5/25, assignment should be hand written on printing pages)**

**Question 1**:  
Explain how the **MSP430FR2355** handles arithmetic overflow and underflow conditions in the **Status Register (SR)**. What are the flags that indicate overflow (V flag) and carry (C flag), and how can these flags be used to detect arithmetic errors during program execution? Provide examples of how an overflow might be handled in a program.

**Key Points**:

* The role of the **V** (overflow) and **C** (carry) flags in the SR.
* The concept of overflow and underflow in the MSP430FR2355's arithmetic operations.
* How to handle overflow conditions in an application (e.g., using conditional branches).

**Question 2**: Write an Assembly program that uses the status flags to make decisions. The program should first perform a subtraction between two registers (R14 and R15). If the result is negative, jump to a label negative\_result and perform an action. If the result is positive, jump to positive\_result and execute another action.

**Requirements**:

* Use the SUB instruction to subtract values.
* Check the status flags (N, Z, C, V) after the operation.
* Use conditional jump instructions based on the flags (JGE, JNE, etc.).

**Question 3**:  
Describe the process of configuring and controlling **digital I/O** pins on the **MSP430FR2355**. Explain the function of the **direction registers (PxDIR)**, **output registers (PxOUT)**, **input registers (PxIN)**, and **resistor enable registers (PxREN)**. How do these registers interact to configure an I/O pin as input or output, and how can internal pull-up or pull-down resistors be enabled?

**Key Points**:

* Explain how I/O pins are configured using control registers.
* The differences between input, output, and bidirectional I/O pin configurations.
* The importance of enabling pull-up or pull-down resistors for input

**Question 4**:  
Describe how an **I/O pin** can be configured on the **MSP430FR2355** to trigger an interrupt. Specifically, explain how to set up a pin for **edge-triggered interrupts** (both rising and falling edges) using the **P1IE** and **P1IES** registers. How does the **interrupt flag (P1IFG)** work in this scenario, and how is it cleared after an interrupt is serviced?

**Key Points**:

* Explain how to configure edge-triggered interrupts on I/O pins.
* The role of **P1IE**, **P1IES**, and **P1IFG** registers in configuring and handling interrupts.
* How to clear interrupt flags to prevent repeated interrupt triggers.