**CSCI 360-1 Assignment 4 Name: Hasnain Attarwala**

**Spring 2019 ABENDs and Dump Reading Z1697740**

**25 points**

**This programming assignment does not require any further coding or documentation than what is provided. The program will ABEND and your task is to learn how to investigate what happened so that you can debug your own Assembler programs in the future. To begin, run the following program on the Marist mainframe using the ASSIST JCL used previously. Be sure you type it or copy it EXACTLY as shown below:**

DUMP1 CSECT

USING DUMP1,15 ESTABLISH REG 15 AS BASE REG

\*

SR 2,2 CLEAR REG 2

\*

LA 3,NUM1 LOAD ADDRESS OF NUM1 INTO REG 3

LA 4,NUM2 LOAD ADDRESS OF NUM2 INTO REG 4

\*

A 2,0(,3) ADD NUM1 TO REG 2

A 2,0(,4) ADD NUM2 TO REG 2

\*

LA 5,SUM LOAD ADDRESS OF SUM INTO REG 5

ST 2,0(,5) STORE THE SUM OF NUM1 & NUM2 AT SUM

\*

XDUMP , SET REG 2 TO 35

\*

LTORG

\*

SUM DC F'0' FULLWORD OF ZERO TO LATER HOLD SUM

\*

NUM1 DC F'1206000000' VARIOUS FULLWORD INTEGER VALUES

NUM2 DC F'972460'

NUM3 DC F'1344335922'

\*

END DUMP1

**Use the resulting ABEND dump output to answer the questions below. Each is worth 2 points except question 11 which is worth 5 points.**

1. Did this error occur (a) while the program was being assembled or (b) when it was being run?

1. What is the address of the next instruction which would have been executed?

ADRESS OF NEXT INSTRUCTION IS **0000030**

1. What is the value of the condition code at the time of the ABEND?

**CC 10, 2(DECIMAL)**

1. What is the length of the instruction that caused the ABEND (number of bytes)?

ILC 10, 2(DECIMAL), ABENDING INSTRUCTION IS 2\*2=**4 BYTEST LONG**

1. What is the address of the instruction that caused the abend?

address of ABDENDING instruction is 0000030-4=**2C**

1. What type of error occurred (number and name)?

**Specification exception, SOC 6**

1. What usually causes this error?

**LOADING / STORING SOMETHING NOT ON BOUNDRY**

1. What does the value in register 2 represent at the time of the ABEND dump?

47F0F02C, 1206972460(BINARY) **SUM OF NUM1 AND NUM2**

1. What instruction needs to be added to fix this ABEND?

**BR 14** RETURN TO CALLER

1. What does the value stored at location counter value 00002C represent?

**Address of the instruction that caused abend aka where we screwed up.**

1. What exactly happened here to cause the ABEND? Be detailed but succinct in your description.

**We were not branching back to caller, we executed storage!**