Computer Architecture
CS 325 - ON40
Department of Physics and Computer Science
Medgar Evers College
Exam 2

Direction: Submit your typed work in the Exams directory of your github repository and/or as an attachment on Google classroom under the Exam02 assessment. All submissions should have their appropriate extensions.

Problem	Maximum Points	Points Earned
1	5	
2	5	
3	5	
4	5	
Total	20	

Instruction commands list for IAS computer

Opcode	Description
0A	Transfer contents from MQ to AC
09	Transfer M(X) to MQ
21	Transfer contents from AC to memory location X
01	Transfer M(X) to AC
05	Add M(X) to AC; put result in AC
06	Subtract M(X) from AC; put result in AC
0B	Multiply M(X) by MQ put must significant bits of result in AC; least significant in MQ
0C	Divide AC by M(X); put quotient in MQ and remainder in AC
14	Multiply AC by 2
15	Divide AC by 2
12	Transfer AC to left address of M(X)
13	Transfer AC to right address of M(X)
0D	Takes next instruction from left half of M(X)
0E	Takes next instruction from right half of M(X)
0F	If AC >= 0, takes next instruction from the left half of M(X)
10	If AC >= 0, takes next instruction from the right half of M(X)

1. Convert each of the following numbers to the requested base. You must show work to receive full points
   a. $4A.C_{16}$ to decimal
   b. 32.75_{10} to binary
   c. 11010.0101_2 to hexadecimal
   d. $AF.AE_{16}$ to binary
   e. 101011.10 to decimal

2. Given a 64-bit processor that has 32-bit instructions in the format of a 2 byte opcode followed by an operand address and memory consists of 64-bit words

   a. What is the maximum directly addressable memory capacity in bytes ?

   b. How many bits are needed for the program counter and the instruction register ?

   c. If the data bus has a width of 32 bits, how many times must the processor access memory for each instruction cycle (read a word) ?

   d. If the width of the control bus is 16 bits, how many lines does the system bus consist of ?

   e. If a module on the system bus wishes to send data to another module, what must it do ?

3. Given the memory contents of the IAS computer shown below,

Address	**Contents**
000	010070F002
001	0000000000
002	0600821007
003	010090500A
004	210090100A
005	0500B2100A
006	0E00000000
007	0000000004
008	0000000001
009	0000000000
00A	0000000002
00B	0000000003

create a trace table of the above program with its header consisting of PC, AC, IR, MBR, and the addresses 007 through 00B. Trace only the first 25 steps and then determine the values of the addresses 007 through 00B when the program terminates. Remember that each line consists of two instructions which are read from left to right. Likewise, before an instruction is executed, it is divided among the IR and MBR in the same step. Furthermore, step 1 is PC equals 000, 007 equals 4, 008 equals 1, 009 equals 0, 00A equals 2 and 00B equals 3. For instance, the trace table of the program below

Address	**Contents**
000	0100205002
001	0000000000
002	000000000F

would be

step	**PC**	**AC**	**IR**	**MBR**	**002**
1	000				F
2			01	002	
3		F			
4			05	002	
5		1E			
6	001				
7			00	000	

4. Write a IAS program that calculates the sum of the cubes of consecutive integers from 1 to n and stores it in address 50. Use the formula
$$\sum_{i=1}^{n} i^2 = \frac{n^2(n+1)^2}{4}$$
Assume that the value of n is stored in address 100. The structure of the program must be like problem 3.

Hint: Store the coeffiecients of the polynomial.

5. **Extra Credit**

A CPU executes multiple programs (or processes); however, they are done one at a time. To avoid having a single program take up too much processor time, programs are scheduled. One scheduling algorithm that is used is called round robin scheduling. This method uses a quota which is a fixed amount of time. Each process, when it is their turn, uses the CPU for their quota. If the process does not terminate before its quota runs out, it is stored and halted, and then, added to the end schedule. Afterwards, the next process in the schedule gets to use the CPU. However, if a process terminates before it quota runs out, it does not return to the schedule.

In the provided cpp file, write the function (`RoundRobin()`) whose header is

```
void RoundRobin(Process procceses[],int n,int quota)
```

where *processes* is an array of type *Process*, which is a structure that consists of a *name* (char) and a *runtime* (int), n is the size of the array, and *quota* is the quota of the round robin scheduler. The function displays the order of the processes terminations when implemented by a round robin scheduler. The name of the processes should be displayed with a space between each of them on their own line. For instance, If *processes* = {(`'A'`,`6`),(`'B'`,`4`),(`'C'`,`3`)} and *quota* = 2, then the function will display `"B C A"`. Furthermore, including any additional libraries to the cpp file will disqualify this extra credit.

Hint: Use a queue.