A1: Architecture (25 points)

Consider these four possible branch predictors:

(A) Static backward taken, forward not taken

(B) 1-bit saturating counter

(C) 2-bit saturating counter

(D) Global predictor (m, n) based on m-bit history and n-bit saturating counter.

Each has advantages and disadvantages in terms of performance and hardware costs.

For each branch pattern below, indicate which type of branch predictor among the four listed above you recommend, in order to get the best performance.

If multiple predictors have the same performance for a given pattern, then your answer should be the predictor that would have the lower hardware cost.

If you choose the global predictor (D), you should also specify your recommended values for m and n.

For the patterns given, there is always one predictor that provides best performance and lower hardware cost.

For each behavior, "T" represents a branch taken, and "N" a branch not taken.

1.1 [6 points]. Behavior:

T T T T T T N T T T T T T T T N T T T T T T T T T N T T T T T T T T T T T N.

Predictor (A, B, C, or D): _____. If you chose D, then $m =$ _____, $n =$ _____.

1.2 [6 points]. In this example, the branch patterns are generated from executions of the following code:

```
for (i=0; i<256; i++) {
  if (i%5 == 0) {
    ...
  }
}
```

The behavior begins:

Branch 1:	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	⋯
Branch 2:	T	N	N	N	N	T	N	N	N	N	T	N	N	N	N	⋯

Predictor (A, B, C, or D): _____. If you chose D, then $m =$ _____, $n =$ _____.

1.3 [6 points]. In this example, there are two branches in a program that are invoked as follows: Branch 1 first, then Branch 2. These exhibit the following behavior:

Branch 1:	T	N	T	T	T	N	N	N	T	N	T	N	T	T	N	T	T	N
Branch 2:	N	T	N	N	N	T	T	T	N	T	N	T	N	N	T	N	N	T

Predictor (A, B, C, or D): _____. If you chose D, then $m =$ _____, $n =$ _____.

1.4 [7 points]. In this example, the branch patterns are generated from executions of the following code:

```
do {
  a = random(0,1);  // a is either 0 or 1
  if (a == 0) { // Branch 1
    ...
  }
  if (a == 1) { // Branch 2
    ...
  }
  i++;
} while (i<10); // Branch 3
```

A possible behavior resulting from executing the code is:

Branch 1:	T	N	T	T	T	N	N	N	T	N
Branch 2:	N	T	N	N	N	T	T	T	N	T
Branch 3:	T	T	T	T	T	T	T	T	T	N

Predictor (A, B, C, or D): _____. If you chose D, then $m =$ _____, $n =$ _____.

A2: Architecture (25 points)

2.1 [8 points]. Indicate which claims are true or false by placing a "T" or "F" in the blank to the left of the claim.

2.1.1: _____. Empirical studies have shown MFLOPS to be a consistent and useful measure of performance.

2.1.2: _____. The performance gain that can be observed by improving some portion of a computer can be estimated fairly accurately using Amdahl's Law.

2.1.3: _____. Synthetic benchmarks generally do a good job in predicting performance for real programs.

2.1.4: _____. A well-designed benchmark will remain valid indefinitely.

2.1.5: _____. Peak performance generally correlates well with observed performance.

2.1.6: _____. The simplest approach to summarizing relative performance is to use total execution time of a program.

2.1.7: _____. MIPS is generally an accurate measure for comparing performance of different computers.

2.1.8: _____. The most important and pervasive principle of computer design is "make the common case fast."

2.2 [6 points]. Four benchmark programs are executed on three computers with the following results:

	Computer A	Computer B	Computer C
Program 1	1	10	20
Program 2	1000	100	20
Program 3	500	1000	50
Program 4	100	800	100

The table shows the execution time in seconds, with $100,000,000$ instructions executed in each of the four programs.

Calculate the MIPS (million instructions per second) rates for each computer for each program, entering the MIPS values into the table below:

	Computer A	Computer B	Computer C
Program 1			
Program 2			
Program 3			
Program 4			

2.3 [6 points]. Now use the MIPS rates computed in **2.2** as your measure of performance to compare the three computers. Calculate the arithmetic and harmonic means of the instruction rates, assuming equal weights for the four programs. Rank the computers based on arithmetic mean and harmonic mean from 1 to 3, where "1" indicates the fastest and "3" indicates the slowest computer, entering your results into the table below:

	Arithmetic mean	Rank	Harmonic mean	Rank
Computer A				
Computer B				
Computer C				

2.4 [5 points]. ____ By looking at the values in **2.3**, which metric better relates to the execution time? (Write letter in blank at left.)

(A) arithmetic mean

(B) harmonic mean

(C) both relate equally well to the execution time

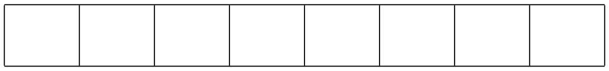
(D) neither relates well to the execution time

A3: Architecture (25 points)

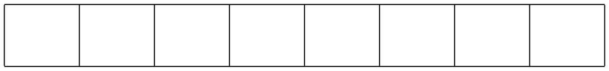
3.1 [6 points]. For each cache mapping policy, place an “X” in all cache blocks into which the memory block with address 12 could be placed.

Each picture represents a cache with 8 blocks, numbered 0 through 7, moving from left to right.

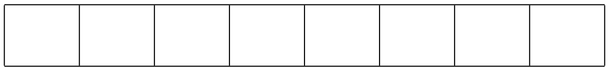
3.1.1. Fully mapped:



3.1.2. Direct mapped:



3.1.3. 4-way associative mapped:



3.2 [7 points]. Consider a cache memory system in which there is a direct-mapped Level 1 (L1) cache and an 8-way set-associative Level 2 (L2) cache, with the following parameters:

Parameter	Value
L1 number of sets	128 sets
L1 block size	4 words
L1 cache access time	2 ns per block
L2 number of sets	128 sets
L2 block size	8 words
L2 cache access time	15 ns per block
Main memory access time	60 ns per word
Main memory size	512M words
L1 cache hit rate	96%
L2 cache hit rate	88%

What is the *effective access time* (also known as the *Average Memory Access Time*) of the system?

Answer: ________________________ ns

3.3 [6 points]. Assume a 32-bit memory address is used to access the L1 cache described in **3.2**. Assume the memory is word-addressed. What is the size of each field of a memory address?

TAG : ______________________ bits

INDEX : ______________________ bits

OFFSET: ______________________ bits

3.4 [6 points]. What is the Average Memory Access Time (AMAT) of a 750 MHz machine with a miss penalty of 20 cycles, a hit time of 2 cycles, and a miss rate of 5%?

AMAT = ______________________

A4: Architecture (25 points)

4.1 [4 points] Match the right dependency relation with the right question, by writing one of the letters in the blank to the left of each question.

Dependency relations:

(A) Data true dependence

(B) Data antidependence

(C) Data output dependence

(D) Control dependence

4.1.1: ____. *On what previous decisions does the execution of a particular instruction depend (in what case will it be reached)?*

4.1.2: ____. *Which instructions subsequently write locations read by the instruction?*

4.1.3: ____. *On the results of which previous instructions does each instruction immediately depend?*

4.1.4: ____. *Which instructions subsequently write locations written by the instruction?*

4.2 [9 points]. Identify all of the dependencies by type in the following code segment. Write your answers as a list of ordered pairs (x, y) of related instructions, where x and y are the numbers that correspond to each instruction, and instruction x depends on instruction y. For example, "$(5, 3)$ $(3, 2)$" is a correctly formatted answer.

```
(1) foo L.D F0, 0(R1)
(2)     ADD.D F2, F0, F3
(3)     L.D F3, 4(R1)
(4)     MUL.D F2, F2, F3
(5)     DADDI R1, R1, #-8
(6)     BNEZ R1, foo
(7)     S.D F2, 8(R1)
```

Data (RAW): ____

Anti (WAR): ____

Out (WAW): ____

4.3 [6 points]. Now assume forwarding hardware is provided and a hazard detection mechanism is used to avoid any remaining data or control hazards by stalling the pipeline. For each instruction, determine in which clock cycle the instruction is safely completed (i.e., the cycle in which its write back stage occurs). Use a 5-stage pipeline. However, since the code deals with floating point operations, stages can have different lengths in terms of number of cycles. Thus assume in-order fetching but out-of-order execution. Moreover, assume the instruction stages are as follows:

Instruction fetch	1 cycle (F)
Instruction decode	1 cycle (D)
Integer execution	1 cycle (X)
FP addition	4 cycles (A1–A4)
FP multiplication	7 cycles (M1–M7)
Memory	1 cycle (M)
Write back	1 cycle (W)
Stalls	each takes 1 cycle (S)

Instruction	Write Back Cycle
(1) `L.D F0, 0(R1)`	
(2) `ADD.D F2, F0, F3`	
(3) `L.D F3, 4(R1)`	
(4) `MUL.D F2, F2, F3`	
(5) `DADDI R1, R1, #-8`	
(6) `BNEZ R1, foo`	
(7) `S.D F2, 8(R1)`	18

4.4 [6 points] What is the throughput (in instructions per cycle) of the pipeline while executing these instructions? Show your work.

IPC = ______________________.