

Problem ST-4 (3 parts)**Pointers, Arrays, and Structs**

Assuming a 32-bit system with 32-bit memory interface and 32-bit addresses, answer the following questions.

Part A Show how the struct definition below maps to memory. Assume it is allocated starting at address 5000. For each variable, draw a box showing its size and position in memory. Label the box with the variable name. Label each element of an array (e.g., Name[0]).

		5000				
		5004				
struct Dog {		5008				
char	Name[6];	5012				
unsigned char	Age;	5016				
float	Weight;	5020				
struct Breed	*MyBreed;	5024				
struct Dog	*Next;					
}						

Part B Suppose the following variables are allocated beginning at address 6000. Complete the table below, listing the value of the expression following this definition.

```
int    A[10] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9};
int    *P = A;
```

A	A + 5	&(A[9])	P - 1
_____	_____	_____	_____
A[9]	P[5]	A == P	P++
_____	_____	_____	_____

Explain what happens if **A** is incremented (e.g. **A++**).

Part C Write the MIPS code implementation of the dynamically allocated array access below in the smallest number of instructions. A pointer to the array (declared below) is stored in \$3. Variables X, Y, Z, and R reside in \$4, \$5 \$6, and \$2, respectively. Modify only \$1 and \$2.

```
int    Array[64][16][32];    /* array declaration */
R = Array[Z][Y][X];          /* implement this */
```

Label	Instruction	Comment