Problem ST-1 (3 parts) **Pointers and Arrays**

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

Suppose the following static variables are allocated in memory beginning at address 5000.

```
int         A[] = {5, 98, 97, 36, 10};
int         C = 25, D = 17, *P = A;
double      H = 3.14;
double      *J = &H;
int         K = 9;
int         *Q = &(A[1]);
```

Part A: Determine the numerical values for the following expressions.

A[5]	**25**	**P[1]**	**98**	**Q[1]**	**97**
***(A+3)**	**36**	**J**	**5032**	**&K**	**5044**

Part B: Can the following statement be implemented, given the declaration of A above? Explain why or why not? **A = A + 1;**

No because A is a constant identifier; it cannot be modified.

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 A, B, and C reside in $4, $5, and $6 respectively. Modify only $1 and $2 and the indexed memory location.

```
int      Array[8][6][32];      /* array declaration */
Array[A][B][C]= 81;            /* implement this */
```

label	instruction	comment
	addi $1, $0, 192	**# LxLy**
	mult $4, $1	**# ALxLy**
	mflo $1	
	sll $2, $5, 5	**# BLx**
	add $2, $2, $1	**# ALxLy + BLx**
	add $2, $2, $6	**# ALxLy + BLx + C**
	sll $2, $2, 2	**# byte offset**
	add $2, $2, $3	**# + base of array**
	addi $1, $0, 81	**# value to store**
	sw $1, 0($2)	**# store 81 at Array[A][B][C]**