LW: Reference and pointer review

Part 1 Questions

Question 1: Given the following code and output, fill out table

```
#include <iostream>
using namespace std;

int main ()
{
    int x = 13;
    int* y = &x;
    int** z = &y;

    cout << x << endl;
    cout << y << endl;
    cout << z << endl;
    cout << &z << endl;
}
```

Output:

```
13
0x7ffe24f3c13c
0x7ffe24f3c130
0x7ffe24f3c128
```

Address	Value
0x7ffe24f3c13c	13
0x7ffe24f3c130	0x7ffe24f3c13c
0x7ffe24f3c128	0x7ffe24f3c130

Question 2: Given table, and provided the following statements, answer questions

```
int x = 27;
int* y = &x;
int** z = &y;
```

Address	Value
0x7ffd7e6af65c	27
0x7ffd7e6af650	0x7ffd7e6af65c
0x7ffd7e6af648	0x7ffd7e6af650

What is the value of *y?

27

What is the value of *z?

0x7ffd7e6af65c

Part 2 Questions

Question 3: Given code and output, fill out table

```
#include<iostream>

using namespace std;

int main(){
	int a_size = 4;
	int *a = new int[a_size];
	for(int i=0; i<a_size; i++){
		a[i] = i;
	}
	cout << *(a+1) << endl;
	cout << *(a+2);
}
```

Output:

1

2

Expression	Result
a[0]	0
a[1]	1
a[2]	2
a[3]	3

Expression	Result
*(a)	0
*(a + 1)	1
*(a + 2)	2
*(a + 3)	3

Question 4: An array's name serves as a pointer to its first element (stores the array's "base" address). What operation(s) does the subscript operator ([]) do when applied as an operator to that pointer? That is, what must happen for the expression a[1] to evaluate to 1 (in question 3)?

Starting from the initial address of the array, the operator adds the index(in this case 1) to the array and gives back the value stored at that address.

Part 3 Question

Question 5: Fill in the missing code (parts #a# through #n#) in functions tripleA and tripleB such that both successfully triple the contents of the argument a. Note the difference in parameter types between the two functions.

```
#include<iostream>
using namespace std;

void tripleA( int**, int*);

int main(){
	int x_size = 4;
	int *x = new int[x_size];
	for(int i=0; i<x_size; ++i){
		x[i] = i;
	}

	tripleA( &x, &x_size   );
}

void tripleA(int **a, int *s){
	for(int i=0; i<( *s ); i++){
		( *a )[i] = ( *a )[i]*3;
	}
}
```

```
#include<iostream>
using namespace std;

void tripleB(  int *&  ,  int &  );

int main(){
	int x_size = 4;
	int *x = new int[x_size];
	for(int i=0; i<x_size; ++i){
		x[i] = i;
	}

	tripleB(x,x_size);
}

void tripleB(int *&a, int &s){
	for(int i=0; i< ( s ); i++){
		( a )[i] = ( a )[i]*3;
	}
}
```