

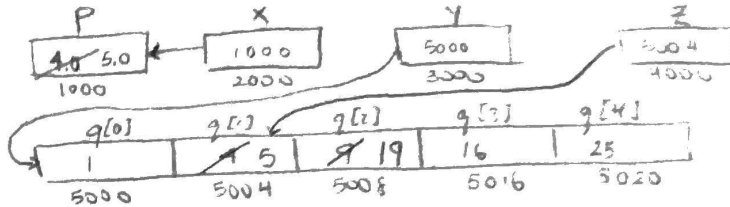
HW 2 – ECE 2620 [Fall 2016]

1. The following code statements appear in sequential order in a program.

```
float p= 4.0, *x;
int q[5]={1,4,9,16,25}, *y, *z; // Assume that the base memory address of array q[] is 5000dec
```

```
x= &p;
p= (*x)+1;

y= q;
z= &q[1];
++(*z);
q[2]= *(y+3)+3;
```



// At this point, write down the values of the following variable expressions. Assume that an int variable occupies 4 bytes in memory, and a float variable occupies 8 bytes

```
// p= 5, *x= 5
// y= 5000, *y= 1
// z= 5004, *z= 5
// q[1]= 5, q[2]= 19
```

```
q= q+2;
```

// Is the above statement legal? If so, what is the new value of q? If not, then why not?

NO, you cannot change the address of an array because X is not a pointer.

```
++y;
```

// Is the above statement legal? If so, what is the new value of y? If not, then why not?

YES 5004

2. Write C++ statements in main() as indicated below.

// Declare a pointer called `p` that is capable of pointing to int-type numbers

```
int *p ;
```

// Create a dynamic array of 10 integers and store its (base) address in the pointer `p` that you had
// declared above

```
p = new int [10];
```

// If `p` points to a non-NULL address, then initialize all array elements to the value 0. If `p` points to a NULL
// address, then return from main() with a return value 1.

```
if (p != NULL) {  
    for (int i=0 ; i<10 ; ++i) {  
        *(p+i) = 0 ;  
    }  
} else {  
    return 1 ;  
}
```

// Remove completely from memory, the above dynamic array

```
delete[] p ;
```