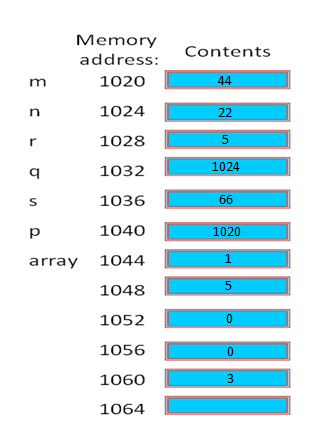
**CS121 Homework – Pointers and 2-D Arrays**

1. (20 Points) Show the contents of the memory locations following execution of the following code:

int m = 44, n, s, r;

int \*p ;

int \*q;

int array[5] = {0};

p = &m;

q = &n;

\*q = 22;

s = \*q + \*p;

array[1] = 5;

\*array = 1;

\*(array + 4) = 3;

r = \*(array + 1);

**Problems 2 – 4 (30 points)**

2. (2 points) True or false?

***a.*** If (x == y) then (&x == &y). T **F**

***b.*** If (x == y) then (\*x == \*y). **T** F

Explain:  
***a.*** Just because two variables have the same value doesn’t mean they always have to be stored at the same address.  
***b.*** If the two pointer variables point to the same address, the de-referenced value of the two pointers must be the same. An address cannot store more than one value at a time.

1. (8 points) What is wrong with the following code?
2. int\* p = &44;

You have to set a pointer to the address of a variable, not the address of an actual value.

1. char c = 'w';

char p = &c;  
The variable p has to be a pointer in order to hold an address.

1. char c = 'w';

char\* p = c;  
A pointer cannot be given a value until it is first given an address to point to (to store the value).

1. float x = 3.14159;

float\* p = &x;

int d = 44;

int\* q = &d;

p = q;  
A floating type pointer cannot store the address of an integer type variable.

1. (20 points) If p and q are pointers to int and n is an int, which of the following are legal:

a. p + q; Legal **Not Legal**

b. p – q; Legal **Not Legal in MS Visual C++**

c. p + n; **Legal** Not Legal

d. p – n; **Legal** Not Legal

e. n + p; **Legal** Not Legal

f.n – q; Legal **Not Legal**  
g. n++;  **Legal**  Not Legal  
h. p++; **Legal**  Not Legal  
i. &q; **Legal**  Not Legal  
j. \*p + n; **Legal** Not Legal