Name:

Pointer Test / Keenan

Assume that we are working on a LITTLE endian processor
unsigned char data[];
Memory Dump (values in Hex)

```
data =  0x0000:  AB CD 12 3F
        0x0004:  33 B5 D3 35
        0x0008:  23 24 01 FE
        0x000C:  CD 33 44 55
        0x0010:  66 03 75 33
        0x0014:  29 55 22 11
        0x0018:  56 88 A9 13
        0x001C:  14 82 68 26
```

```
unsigned char  *p;  // char are 8-bits wide
unsigned int   *r;  // ints are 32-bits wide
unsigned short *s;  // shorts are 16-bits wide
```

```
p = &data[0];                          Expected output
printf("%x\n",  *(p+3)  );     1)_____________________________
printf("%x\n",  *(p+5)  );     2)_____________________________
p = p + 12;
printf("%x\n",   *(p)  );      3)_____________________________
printf("%x\n",   p[2]  );      4)_____________________________
printf("%x\n",   *p++  );      5)_____________________________
p += 6;
printf("%x\n",   *--p  );      6)_____________________________
printf("%x\n",   p[5]  );      7)_____________________________
p = p + 2;
printf("%x\n",   *p++  );      8)_____________________________
printf("%x\n",  *(p+3) );      9)_____________________________
p = 5 + p;
printf("%x\n",  *(p++) );     10)_____________________________
printf("%x\n",  *(--p) );     11)_____________________________
```

```
data =  0x0000:  AB CD 12 3F
        0x0004:  33 B5 D3 35
        0x0008:  23 24 01 FE
        0x000C:  CD 33 44 55
        0x0010:  66 03 75 33
        0x0014:  29 55 22 11
        0x0018:  56 88 A9 13
        0x001C:  14 82 68 26

r = (unsigned int *)&data[0]
printf("%x\n",   *(r)  );        12)_____
printf("%x\n",  *(r+5) );        13)_____
r++;
printf("%x\n",   *r++  );        14)_____
r = r + 2;
printf("%x\n",   r[2]  );        15)_____
r = r + 1;
printf("%x\n",   r[0]  );        16)_____

s = (unsigned short *) r;
printf("%x\n",   s[-2] );        17)_____
s = s - 3;
printf("%x\n",   s[2]  );        18)_____
s += 5;
printf("%x\n",  *(s+3) );        19)_____
printf("%x\n",   *(s)  );        20)_____
p = (unsigned char *) s;
printf("%x\n",  *(p+3) );        21)_____
p += 5;
printf("%x\n",  p[-9] );         22)_____
--p;
printf("%x\n",   p[0] );         23)_____
```