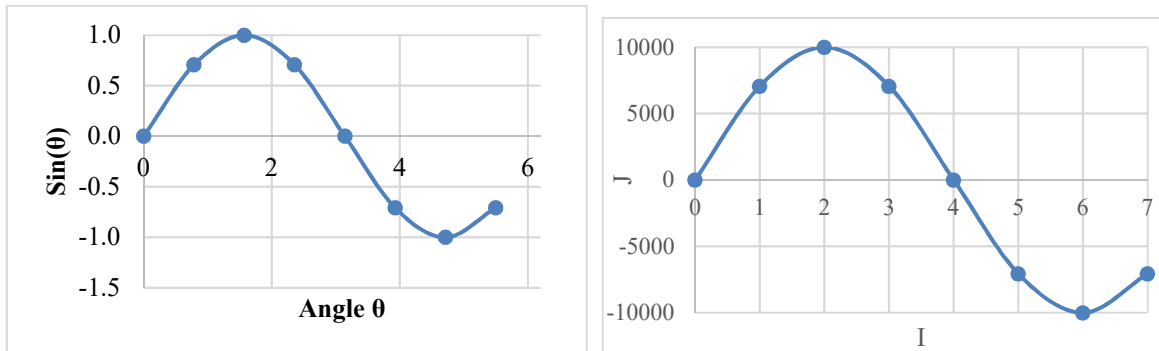


(25 pts) Q1) Write an **assembly** function that implements the $\sin(\theta)$ function. Your function will convert a **3-bit unsigned input, I** into a **16-bit signed output, J**. Let **I** be the integer portion of an angle in fixed point with resolution of $\pi/4$. We will restrict the input angle to $0 \leq \theta \leq 7\pi/4$. Therefore, the 3-bit input **I** will vary from $0 \leq I \leq 7$. In particular,

$$\theta = I * (\pi/4)$$

The output is a fixed point number with a resolution of 0.0001. Let **J** the integer portion of $\sin(\theta)$.

$$\sin(\theta) = J * 0.0001$$



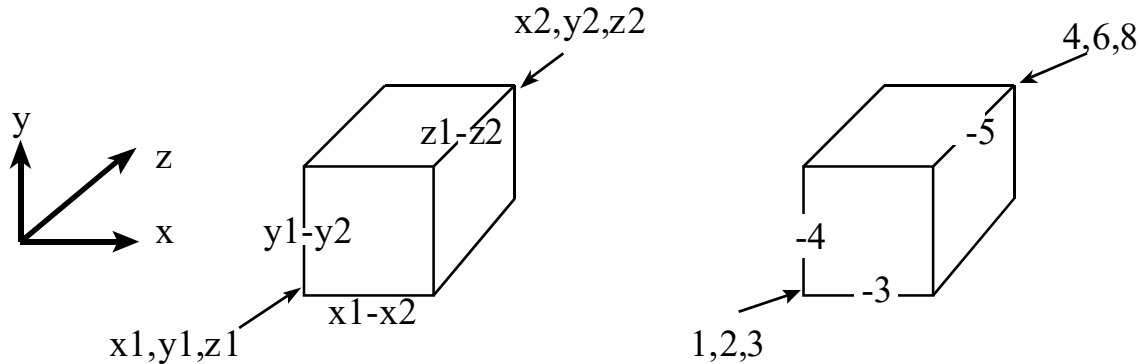
See the software comments for this data in numerical form.

Hint: Create a 16-bit signed array in ROM, and then index into the array within your **Sine** function.

I	J
0	0
1	7071
2	10000
3	7071
4	0
5	-7071
6	-10000
7	-7071

(30pts) Q2) Write a **C** function that finds the largest even number in a variable length array. Data are 8-bit signed numbers, with -128 as the termination code. See the comments in the program for specific details of the array and error conditions.

Both Q3 and Q4 use the same data structure. Cubes are aligned with the x-y-z axes. A cube is defined by a color and two points. There are five possible colors. $(x1, y1, z1)$ is the coordinates of one corner, and $(x2, y2, z2)$ is the coordinates of the opposite corner. *The distance $x1-x2$ is signed.*



```
struct cube{
    unsigned short color; // 16-bit color, 5 possibilities
    short x1,y1,z1;       // one corner
    short x2,y2,z2;       // opposite corner
};
typedef struct cube cube_t;
```

(30 pts) Q3) Write a function in C that finds the volume of a cube. Volumes can be positive or negative. The signed volume is $(x1-x2)$ times $(y1-y2)$ times $(z1-z2)$. See the comments in the program for specific details, some example input/output, and error conditions.

(15 pts) Part d) Write a function in C that searches an array of cubes for two cubes of the same red color and the same volume. The input to the function is a pointer to an array of cubes, and the number of cubes in the array. The output is the largest volume of two matching cubes (both red and both having the same volume). You are allowed to call the function you wrote in part c) but are not required to call it. See the comments in the program for specific details, some example input/output, and error conditions.

You may not use any functions from any C libraries other than **stdint.h**.