1. vals is stored at 0x602010. partial\_sums is stored at 0x602030. I got these using the commands “print vals” and “print partial sums”
2. I used x/10dw vals. The values are 1,2,4,8,16,32,64,128,256,512
3. I used x/10dw vals and x/10dw partial\_sums. The values in vals are 1,2,4,8,16,32,64,128,0,0. The values in partial\_sums are all 0. This is not what they are supposed to be.
4. These addresses are 32 bytes away from each other. This is enough to store 8 integers, which makes sense because vals now only contains 8 integers (the last 2 got removed and are now zero).
5. The values in partial\_sums are 1,3,7,15,31,63,127,255,256,259. These are the values that are printed. However, the first two values in partial sums overwrote the last two values in vals, which is why the last two values in partial sums are incorrect (as they are 255+1=256 and 256+3=259 instead of adding the proper power of two).
6. The content of the partial arrays does not change. However, between lines 24 and 34, the values in vals get overridden when the first two elements in partial sums are initialized to 0. This is because not enough memory was allocated to the vals array.