

1. For the first part, I showed the argv address on the right and its actual value in the middle using a custom address formatter which displays the addresses correctly. This formatter was also used for the second part where I went through each item in the argv list. After this, I figured out the closest address to the first item in argv which is also a multiple of 8 but made sure this would be before the address of the first item. Then I went through the last element and found the address which comes at either the end of the last item or is a multiple of 8 again which would be where I would know to stop printing values. With both the start and end, I made a function which takes each line of the address and prints it out with each element in it until I reached the end.
2. For the second problem, I first made the optstring which takes care of the options that are used but also the options that need arguments. I also used ':' at the beginning so whenever an option does not have an argument, I have a special case to handle that situation. Also, sometimes an option can be interpreted as an argument to another option if the previous one was missing an argument so I also added an if statement to take care of that situation. At the end, I go through all of the operands.
3. First, I go through the envp array to get the total number of variables in it and then pass in the count and the array in my bubblesort function. The function takes care of sorting the array and for each comparison, it makes a copy of the line so it does not set the delimiter to null when I run the strtok on it. I also free both copies at the end to make sure there are no memory leaks.