Introduction to Parallel Sorting:

Count sorts may be parallelized in 2 ways:

1. Data division
    1. Divide the data among the processors (ie scatter)
    2. Each processor counts the data it owns
    3. Sum reduction across the processors to get the final count
2. Bucket division
    1. divide the buckets across the processors
    2. broadcast (chunks of the data)
    3. each processor updates the buckets that it owns
    4. Gather at the end to get the final collection

Merge Sorts

The idea for parallel merge sorting is quite similar the sequential version:

1. Divide the data among the processors
2. Sort the data on the individual processors, using your favorite sequential sort.
3. Merge the various

Note that the merge operation is a reduction operation!

merge([x1,x2,...],[y1,y2,...]) = [x1,y1,x2,y2,...]

Reference: *Parallel Integer Sorting.pdf by Andrew Tridgell*
Please at least read Chapter 2.4. Here contains all the information introduce different merging mehod.

SPI bus in C8051F020:
The Serial Peripheral Interface (SPI0) provides access to a four-wire, full-duplex, serial bus. SPI0 may operate as a master or a slave, and supports the connection of multiple slaves and masters on the same bus. A slave-select input (NSS) is included in the SPI0 interface to select SPI0 as a slave; additional general purpose port I/O can be used as slave-select outputs when SPI0 is operating as a master. The maximum data transfer rate (bits/sec) must be less that 1/10 the system clock frequency.

The four signals used by SPI0 (MOSI, MISO, SCK, NSS).

Details please check P197 in C8051F020 datasheet.

Your Task:
Merge Sort with Unbalanced Merging(devide the 500 input into two arrays with 400 and 100, pass the data to another MCU use SPI, after sort separately, merge sort via SPI);

Suggestion for getting start:
First, get familiar with the sorting algorithm. Try it from small size of array to large size of array.

Second, start to implement the SPI bus communication. This require wires, which will be available to you soon. When the wires are available, you are going to be contacted. Before this, please be fully prepared.

Then, you can play with the algorithm implementation. You may need this virtual tool to help you output final result, or even input from the computer.

Your input:

```
unsigned char input[500] = {92  134 106 143 104 151 161 203 121 38  94  188 84
    171 135 191 46  121 78  254 165 189 84  110 118 177 118 160 42  114 94  62
    151 142 131 73  178 145 117 131 118 55  117 94  88  80  107 44  171 152
    129 128 95  173 124 99  187 120 104 117 93  82  238 200 143 76  93  122
    163 73  30  68  144 146 149 124 137 109 166 71  149 93  115 153 174 82
    184 158 127 121 120 117 131 132 165 195 150 121 156 137 86  170 143 135
    152 141 89  123 123 107 202 92  109 99  79  121 118 195 119 84  198 182
    120 65  111 123 141 118 149 146 76  89  98  108 116 130 0   110 183 84
    170 145 128 137 63  126 198 134 131 98  128 140 148 114 119 216 33  225
    144 172 58  104 118 148 58  150 78  132 158 144 176 173 102 141 89  73
    169 130 127 169 155 145 183 169 140 100 102 181 61  129 46  173 166 130
    127 23  154 36  30  133 89  147 159 166 100 149 134 165 153 168 124 123
    173 39  108 75  113 157 165 86  109 135 117 143 147 90  122 38  179 103 78
    119 68  129 106 223 178 23  148 70  119 137 162 118 197 109 144 158 133
    167 143 96  52  209 104 134 154 134 91  110 124 193 93  163 143 120 84
    117 126 67  138 94  126 144 91  117 145 51  174 233 171 116 148 85  210
    170 163 92  143 106 116 105 86  91  121 57  156 125 160 141 151 66  86
    110 134 178 117 184 150 180 135 101 66  136 165 117 106 116 83  109 122
    132 127 156 134 207 143 207 99  152 118 155 155 36  73  68  147 192 116
    164 153 85  147 97  194 128 200 111 155 127 43  88  156 127 82  103 140 87
    171 102 207 83  138 65  99  104 147 170 142 114 164 164 135 154 147 87
    162 101 104 137 116 124 155 174 121 144 119 139 148 103 141 155 134 204
    104 98  55  169 167 126 168 137 142 134 148 134 249 80  50  81  83  111
    122 120 153 146 162 206 182 75  30  168 51  132 131 225 127 108 140 140
    133 104 77  143 72  85  187 112 124 168 117 174 115 173 157 120 93  85
    118 111 112 172 117 179 107 171 107 137 171 112 111 215 170 111 157 114
    160 190 61  174 192 132 204 136 77  36  115 160 143 147 105 136 59  97  95
    152 129 80   129 100 101 167 134 147 167 137 153 159 180 150};
```

Reference: *VirtualToolsGuide*