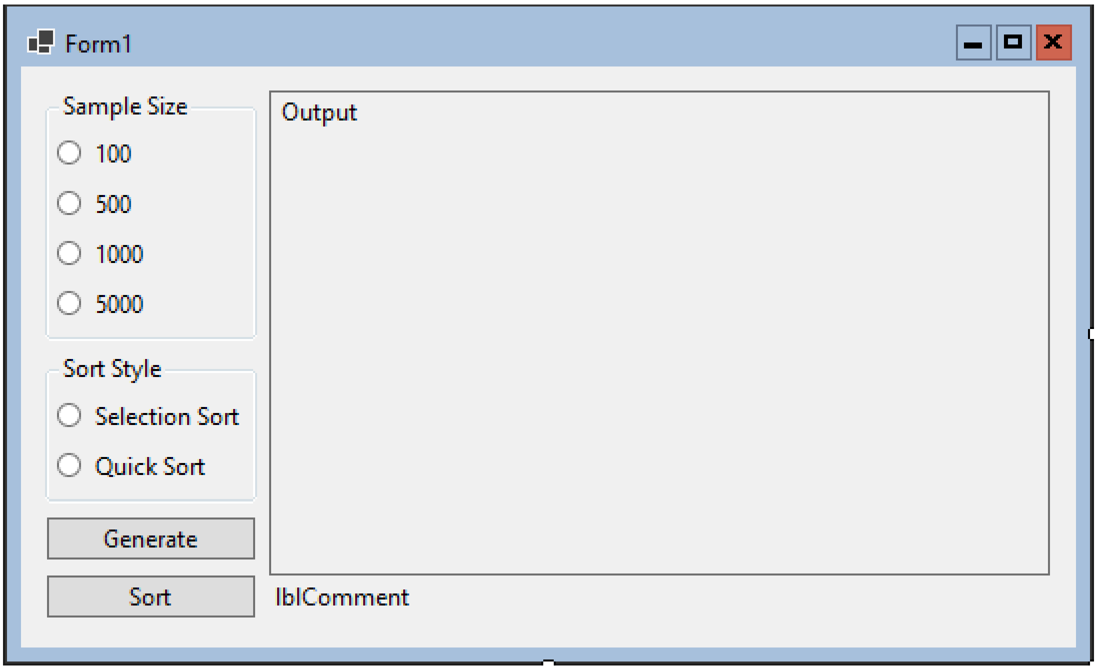
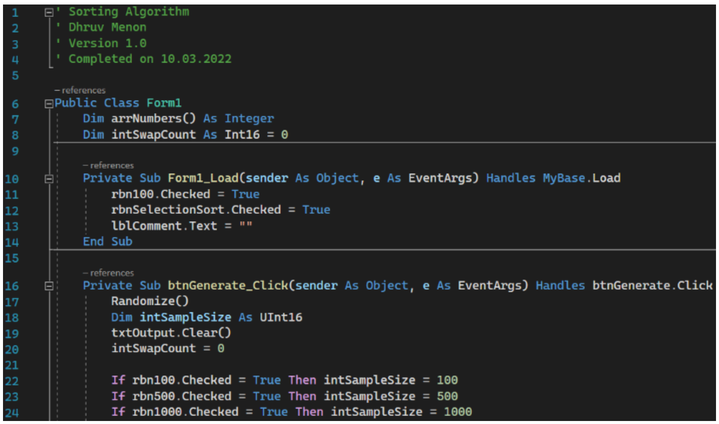
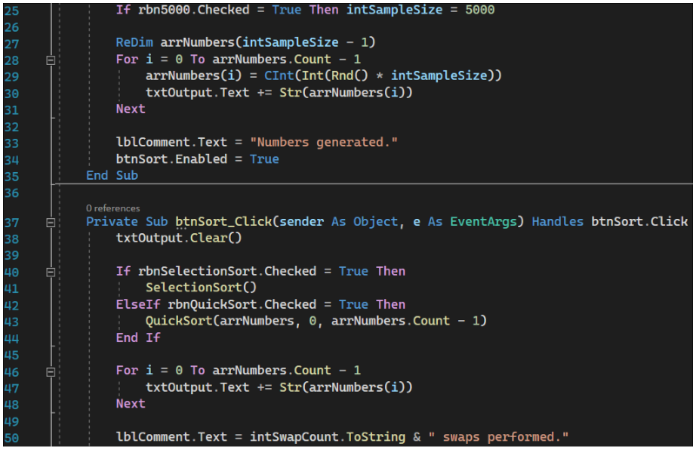
Computing – Software Development: Sort Algorithm

Dhruv Menon

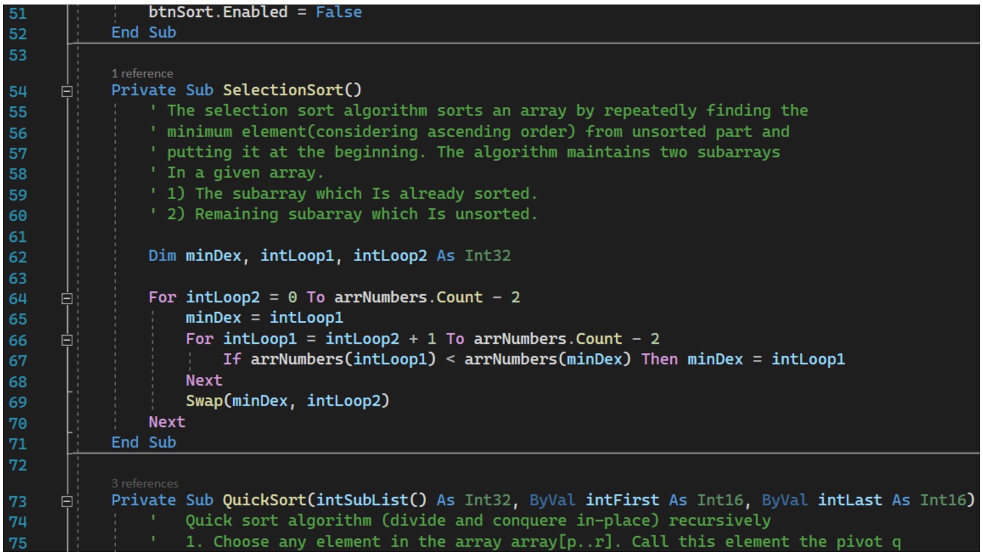
Form:



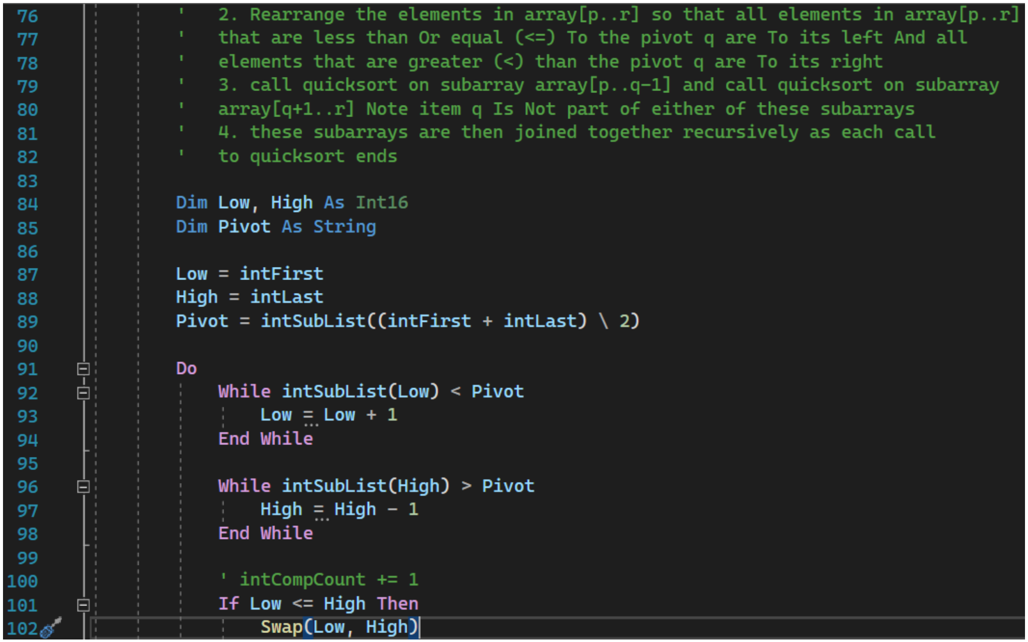
Code:

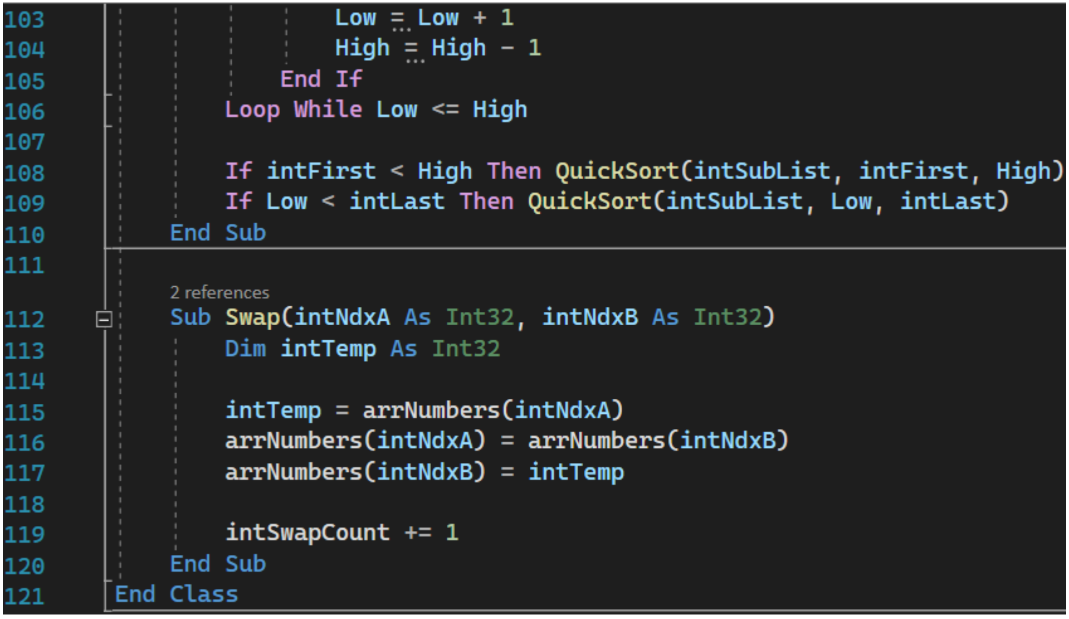
```
' Sorting Algorithm
' Dhruv Menon
' Version 1.0
' Completed on 10.03.2022

– references
Public Class Form1
    Dim arrNumbers() As Integer
    Dim intSwapCount As Int16 = 0

    – references
    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        rbn100.Checked = True
        rbnSelectionSort.Checked = True
        lblComment.Text = ""
    End Sub

    – references
    Private Sub btnGenerate_Click(sender As Object, e As EventArgs) Handles btnGenerate.Click
        Randomize()
        Dim intSampleSize As UInt16
        txtOutput.Clear()
        intSwapCount = 0

        If rbn100.Checked = True Then intSampleSize = 100
        If rbn500.Checked = True Then intSampleSize = 500
        If rbn1000.Checked = True Then intSampleSize = 1000
        If rbn5000.Checked = True Then intSampleSize = 5000

        ReDim arrNumbers(intSampleSize - 1)
        For i = 0 To arrNumbers.Count - 1
            arrNumbers(i) = CInt(Int(Rnd() * intSampleSize))
            txtOutput.Text += Str(arrNumbers(i))
        Next

        lblComment.Text = "Numbers generated."
        btnSort.Enabled = True
    End Sub

    0 references
    Private Sub btnSort_Click(sender As Object, e As EventArgs) Handles btnSort.Click
        txtOutput.Clear()

        If rbnSelectionSort.Checked = True Then
            SelectionSort()
        ElseIf rbnQuickSort.Checked = True Then
            QuickSort(arrNumbers, 0, arrNumbers.Count - 1)
        End If

        For i = 0 To arrNumbers.Count - 1
            txtOutput.Text += Str(arrNumbers(i))
        Next

        lblComment.Text = intSwapCount.ToString & " swaps performed."
```



```
            btnSort.Enabled = False
        End Sub

        1 reference
        Private Sub SelectionSort()
            ' The selection sort algorithm sorts an array by repeatedly finding the
            ' minimum element(considering ascending order) from unsorted part and
            ' putting it at the beginning. The algorithm maintains two subarrays
            ' In a given array.
            ' 1) The subarray which Is already sorted.
            ' 2) Remaining subarray which Is unsorted.

            Dim minDex, intLoop1, intLoop2 As Int32

            For intLoop2 = 0 To arrNumbers.Count - 2
                minDex = intLoop1
                For intLoop1 = intLoop2 + 1 To arrNumbers.Count - 2
                    If arrNumbers(intLoop1) < arrNumbers(minDex) Then minDex = intLoop1
                Next
                Swap(minDex, intLoop2)
            Next
        End Sub

        3 references
        Private Sub QuickSort(intSubList() As Int32, ByVal intFirst As Int16, ByVal intLast As Int16)
            '   Quick sort algorithm (divide and conquere in-place) recursively
            '   1. Choose any element in the array array[p..r]. Call this element the pivot q
```

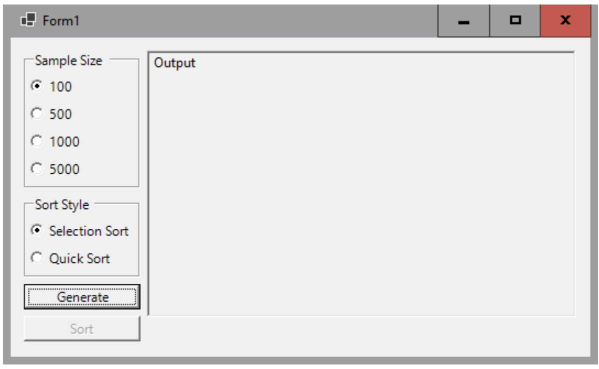


```
            '   2. Rearrange the elements in array[p..r] so that all elements in array[p..r]
            '   that are less than Or equal (<=) To the pivot q are To its left And all
            '   elements that are greater (<) than the pivot q are To its right
            '   3. call quicksort on subarray array[p..q-1] and call quicksort on subarray
            '   array[q+1..r] Note item q Is Not part of either of these subarrays
            '   4. these subarrays are then joined together recursively as each call
            '   to quicksort ends

            Dim Low, High As Int16
            Dim Pivot As String

            Low = intFirst
            High = intLast
            Pivot = intSubList((intFirst + intLast) \ 2)

            Do
                While intSubList(Low) < Pivot
                    Low = Low + 1
                End While

                While intSubList(High) > Pivot
                    High = High - 1
                End While

                ' intCompCount += 1
                If Low <= High Then
                    Swap(Low, High)
```



```
                    Low = Low + 1
                    High = High - 1
                End If
            Loop While Low <= High

            If intFirst < High Then QuickSort(intSubList, intFirst, High)
            If Low < intLast Then QuickSort(intSubList, Low, intLast)
        End Sub

        2 references
        Sub Swap(intNdxA As Int32, intNdxB As Int32)
            Dim intTemp As Int32

            intTemp = arrNumbers(intNdxA)
            arrNumbers(intNdxA) = arrNumbers(intNdxB)
            arrNumbers(intNdxB) = intTemp

            intSwapCount += 1
        End Sub
    End Class
```

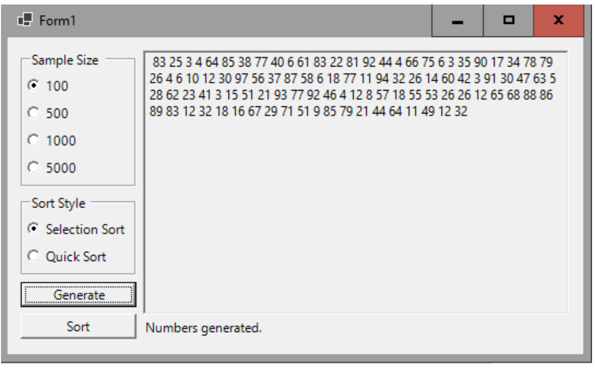
Valid input (radio buttons selected, generate button clicked)/output (numbers generated):



Sample Size
● 100
○ 500
○ 1000
○ 5000

Sort Style
● Selection Sort
○ Quick Sort

Output



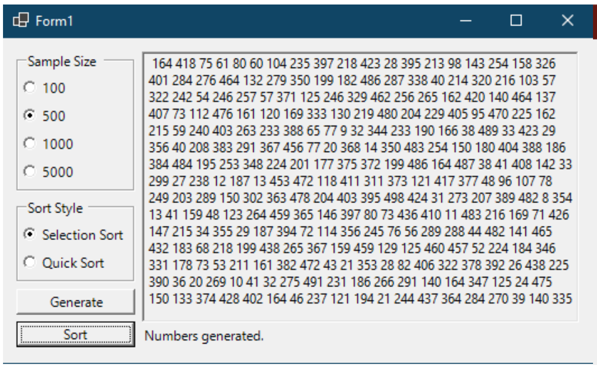
Sample Size
● 100
○ 500
○ 1000
○ 5000

Sort Style
● Selection Sort
○ Quick Sort

```
 83 25 3 4 64 85 38 77 40 6 61 83 22 81 92 44 4 66 75 6 3 35 90 17 34 78 79
26 4 6 10 12 30 97 56 37 87 58 6 18 77 11 94 32 26 14 60 42 3 91 30 47 63 5
28 62 23 41 3 15 51 21 93 77 92 46 4 12 8 57 18 55 53 26 26 12 65 68 88 86
89 83 12 32 18 16 67 29 71 51 9 85 79 21 44 64 11 49 12 32
```

Numbers generated.

Valid input (sort selected, sort button clicked)/output (numbers sorted):

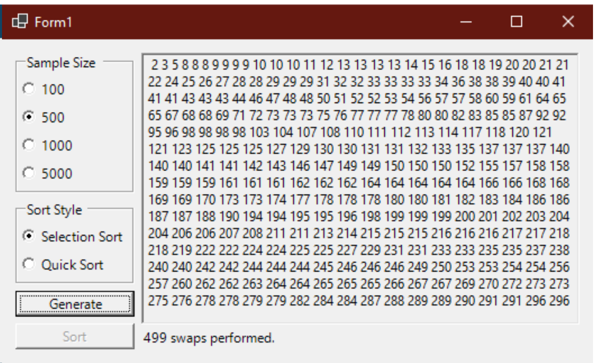


Sample Size
○ 100
● 500
○ 1000
○ 5000

Sort Style
● Selection Sort
○ Quick Sort

```
 164 418 75 61 80 60 104 235 397 218 423 28 395 213 98 143 254 158 326
401 284 276 464 132 279 350 199 182 486 287 338 40 214 320 216 103 57
322 242 54 246 257 57 371 125 246 329 462 256 265 162 420 140 464 137
407 73 112 476 161 120 169 333 130 219 480 204 229 405 95 470 225 162
215 59 240 403 263 233 388 65 77 9 32 344 233 190 166 38 489 33 423 29
356 40 208 383 291 367 456 77 20 368 14 350 483 254 150 180 404 388 186
384 484 195 253 348 224 201 177 375 372 199 486 164 487 38 41 408 142 33
299 27 238 12 187 13 453 472 118 411 311 373 121 417 377 48 96 107 78
249 203 289 150 302 363 478 204 403 395 498 424 31 273 207 389 482 8 354
13 41 159 48 123 264 459 365 146 397 80 73 436 410 11 483 216 169 71 426
147 215 34 355 29 187 394 72 114 356 245 76 56 289 288 44 482 141 465
432 183 68 218 199 438 265 367 159 459 129 125 460 457 52 224 184 346
331 178 73 53 211 161 382 472 43 21 353 28 82 406 322 378 392 26 438 225
390 36 20 269 10 41 32 275 491 231 186 266 291 140 164 347 125 24 475
150 133 374 428 402 164 46 237 121 194 21 244 437 364 284 270 39 140 335
```

Numbers generated.

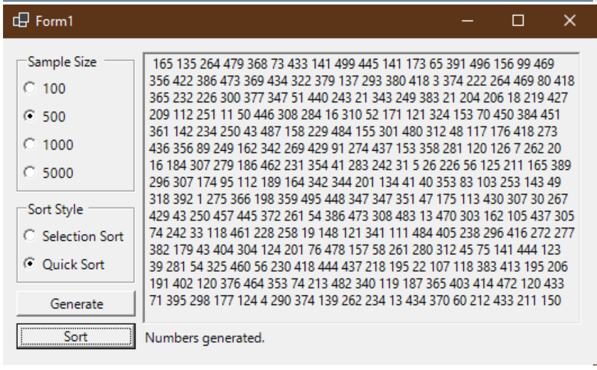


Sample Size
○ 100
● 500
○ 1000
○ 5000

Sort Style
● Selection Sort
○ Quick Sort

```
 2 3 5 8 8 8 9 9 9 9 10 10 10 11 12 13 13 13 13 14 15 16 18 18 19 20 20 21 21
22 24 25 26 27 28 28 29 29 29 31 32 32 33 33 33 33 34 36 38 38 39 40 40 41
41 41 43 43 43 44 46 47 48 48 50 51 52 52 53 54 56 57 57 58 60 59 61 64 65
65 67 68 68 69 71 72 73 73 73 75 76 77 77 77 78 80 80 82 83 85 85 87 92 92
95 96 98 98 98 98 103 104 107 108 110 111 112 113 114 117 118 120 121
121 123 125 125 125 127 129 130 130 131 131 132 133 135 137 137 137 140
140 140 141 141 142 143 146 147 149 149 150 150 150 152 155 157 158 158
159 159 159 161 161 161 162 162 162 164 164 164 164 164 166 166 168 168
169 169 170 173 173 174 177 178 178 178 180 180 181 182 183 184 186 186
187 187 188 190 194 194 195 195 196 198 199 199 199 200 201 202 203 204
204 206 206 207 208 211 211 213 214 215 215 215 216 216 216 217 217 218
218 219 222 222 224 224 225 225 227 229 231 231 233 233 235 235 237 238
240 240 242 242 244 244 244 245 246 246 246 249 250 253 253 254 254 256
257 260 262 262 263 264 264 265 265 265 266 267 267 269 270 272 273 273
275 276 278 278 279 279 282 284 284 287 288 289 289 290 291 291 296 296
```

499 swaps performed.

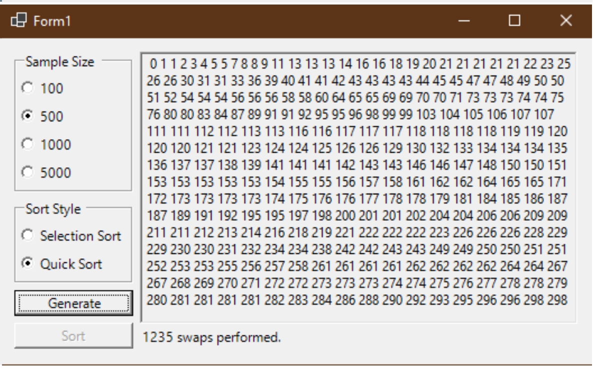


Sample Size
○ 100
● 500
○ 1000
○ 5000

Sort Style
○ Selection Sort
● Quick Sort

```
 165 135 264 479 368 73 433 141 499 445 141 173 65 391 496 156 99 469
356 422 386 473 369 434 322 379 137 293 380 418 3 374 222 264 469 80 418
365 232 226 300 377 347 51 440 243 21 343 249 383 21 204 206 18 219 427
209 112 251 11 50 446 308 284 16 310 52 171 121 324 153 70 450 384 451
361 142 234 250 43 487 158 229 484 155 301 480 312 48 117 176 418 273
436 356 89 249 162 342 269 429 91 274 437 153 358 281 120 126 7 262 20
16 184 307 279 186 462 231 354 41 283 242 31 5 26 226 56 125 211 165 389
296 307 174 95 112 189 164 342 344 201 134 41 40 353 83 103 253 143 49
318 392 1 275 366 198 359 495 448 347 347 351 47 175 113 430 307 30 267
429 43 250 457 445 372 261 54 386 473 308 483 13 470 303 162 105 437 305
74 242 33 118 461 228 258 19 148 121 341 111 484 405 238 296 416 272 277
382 179 43 404 304 124 201 76 478 157 58 261 280 312 45 75 141 444 123
39 281 54 325 460 56 230 418 444 437 218 195 22 107 118 383 413 195 206
191 402 120 376 464 353 74 213 482 340 119 187 365 403 414 472 120 433
71 395 298 177 124 4 290 374 139 262 234 13 434 370 60 212 433 211 150
```

Numbers generated.



Sample Size
○ 100
● 500
○ 1000
○ 5000

Sort Style
○ Selection Sort
● Quick Sort

```
 0 1 1 2 3 4 5 5 7 8 8 9 11 13 13 13 14 16 16 18 19 20 21 21 21 21 21 22 23 25
26 26 30 31 31 33 36 39 40 41 41 42 43 43 43 43 44 45 45 47 47 48 49 50 50
51 52 54 54 54 56 56 56 58 58 60 64 65 65 69 69 70 70 71 73 73 73 74 74 75
76 80 80 83 84 87 89 91 91 92 95 95 96 98 99 99 103 104 105 106 107 107
111 111 112 112 113 113 116 116 117 117 117 118 118 118 118 119 119 120
120 120 121 121 123 124 124 125 126 126 129 130 132 133 134 134 134 135
136 137 137 138 139 141 141 141 142 143 143 146 146 147 148 150 150 151
153 153 153 153 153 154 155 155 156 157 158 161 162 162 164 165 165 171
172 173 173 173 173 174 175 176 176 177 178 178 179 181 184 185 186 187
187 189 191 192 195 195 197 198 200 201 201 202 204 204 206 206 209 209
211 211 212 213 214 216 218 219 221 222 222 222 223 226 226 226 228 229
229 230 230 231 232 234 234 238 242 242 243 243 249 249 250 250 251 251
252 253 253 255 256 257 258 261 261 261 261 262 262 262 262 264 264 267
267 268 269 270 271 272 272 273 273 273 274 274 275 276 277 278 278 279
280 281 281 281 281 282 283 284 286 288 290 292 293 295 296 296 298 298
```

1235 swaps performed.

[No invalid input/output]