User Documentation

Selection-Sort Visualizer

The software is Developed on Microsoft Visual studio 2013 using Visual basic. For best experience we advise users to follow some basic guidelines:

There will be three types of sorting available:

Enter less than 17 numbers for seeing visualisation. If you enter more than 17 numbers, we won't show you any visualisation but will just sort the array.

1. Numerical Sorting: - In this sorting we will sort only numbers, for visualization to work best, ***enter only decimals from –999 to +999.***

   NOTE: If you enter more than 999 or less than –999, we will not show you the visualisation but will show you the sorted array.

2. Alphabetical Sorting: - In this type of sorting we take the input as strings. In this type of sorting you can only input numbers and alphabets. ***You can only input strings less than the length of 5.*** Their sorting precedence is as follows: -
   a. 0-9 are the smallest characters
   b. Precedence is A,a,B,b,C,c.....,Z,z

   NOTE: If you enter strings of length more than 5, we will show you just the sorted array and not the visualisation.

3. Lexicographic Sorting: - This type of sorting is based on ascii values. You can only input numbers and alphabets in this field. ***You can only input strings less than the length of five***. The sorting precedence is as follows: -
   a. 0-9 are the smallest characters
   b. A-Z are the next larger characters
   c. a-z are the next larger characters.

   NOTE: If you enter strings of length more than 5, we will show you just the sorted array and not the visualisation.

   LINK TO UNDERSTAND ASCII VALUES https://www.computerhope.com/jargon/a/ascii.htm

If user won't follow guidelines 1,2 or 3, the program will throw a proper warning.

The algorithm used for sorting is called ***SELECTION SORT.***

Algorithm Description

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.

Steps of Algorithm:

1. Find the smallest number. Swap it with the first element of the array.
2. Find the second-smallest number. Swap it with the second element of the array.
3. Find the third-smallest number. Swap it with the second element of the array.
4. Repeat finding the next-smallest element and swapping it into the correct position until the array is sorted.

Time Complexity: O(n^2)

Auxiliary Space: O(1)

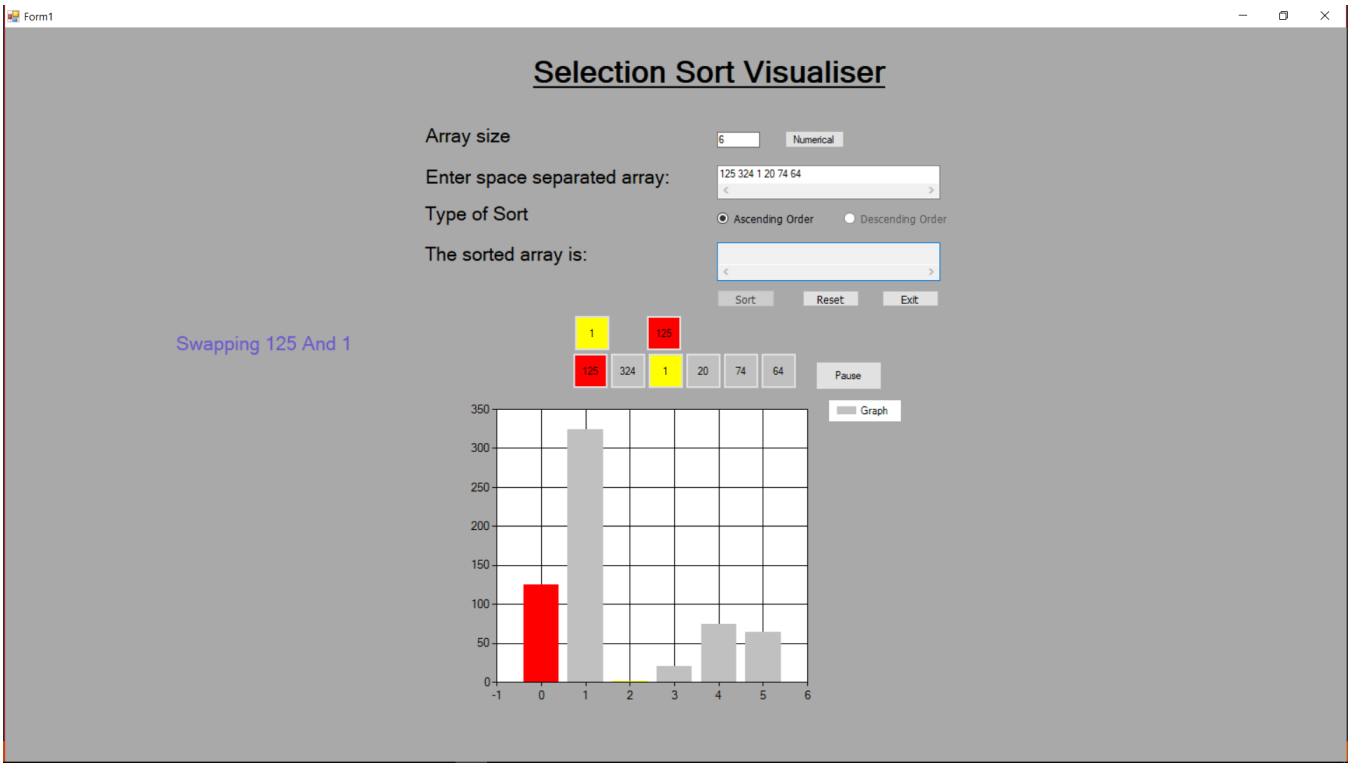
Some useful instructions on how to use the program

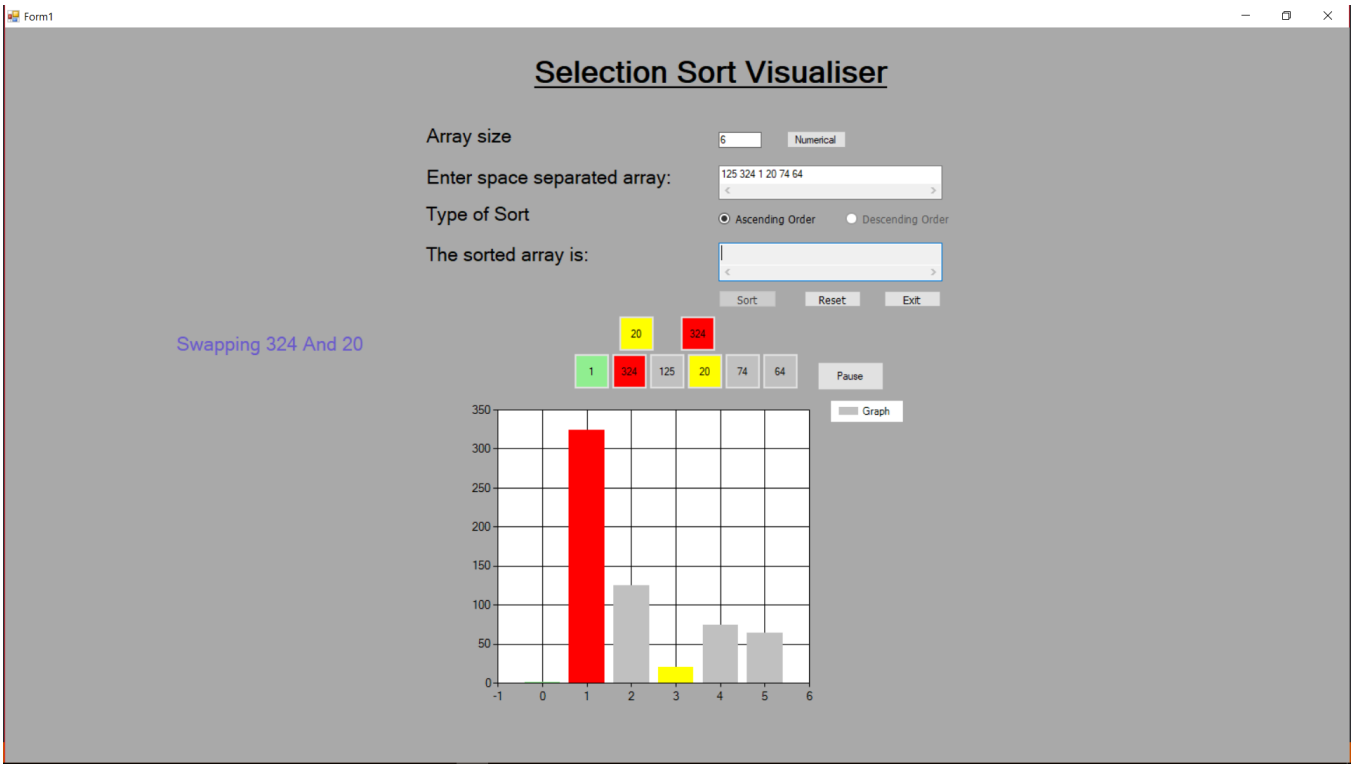
User has to provide the size of array("**INTEGER**"), Elements he/she wants to sort in "**SPACE SEPERATED DECIMALS**" and has to select the order either Ascending or Descending order .By default the order is ascending order.

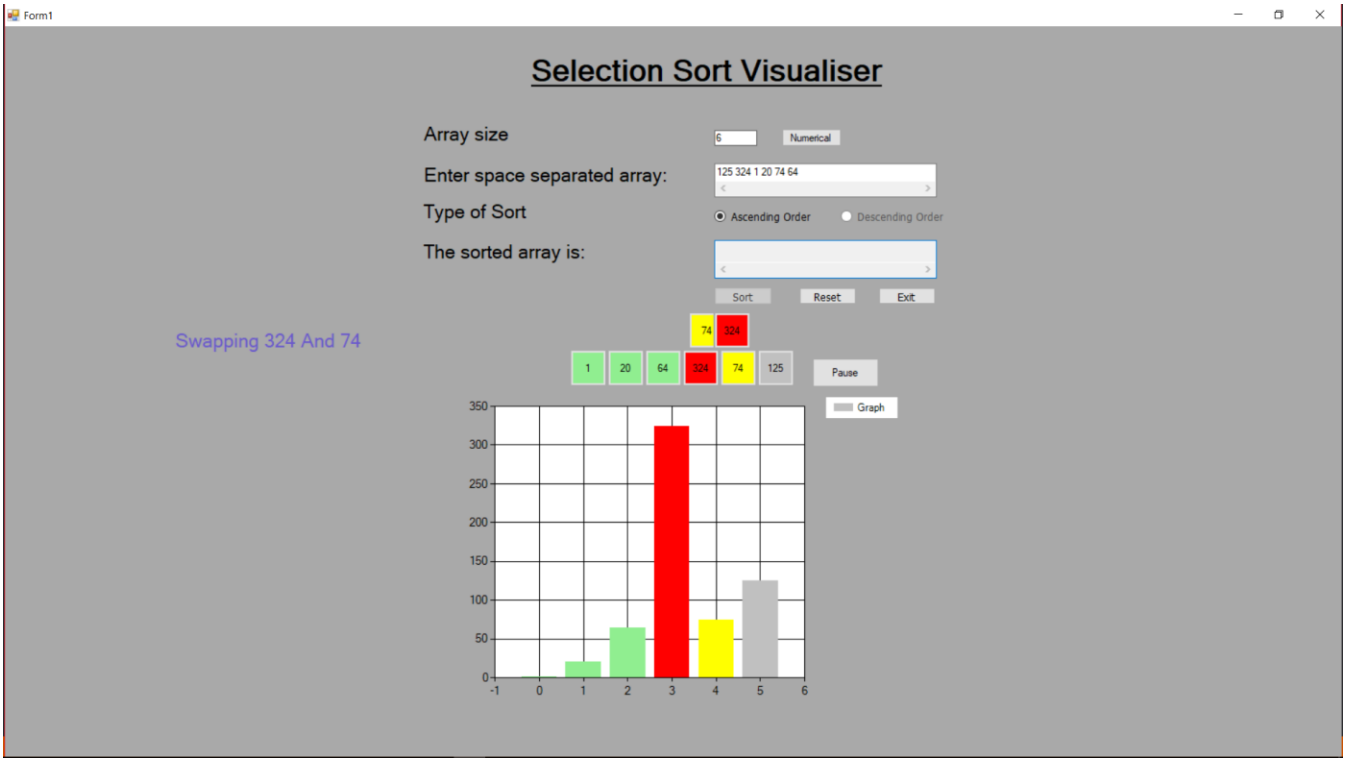
User should check the above conditions properly before hitting the **SORT** button, as if any one of the condition fails will lead to wrong output or **RESET** or Termination of the program.

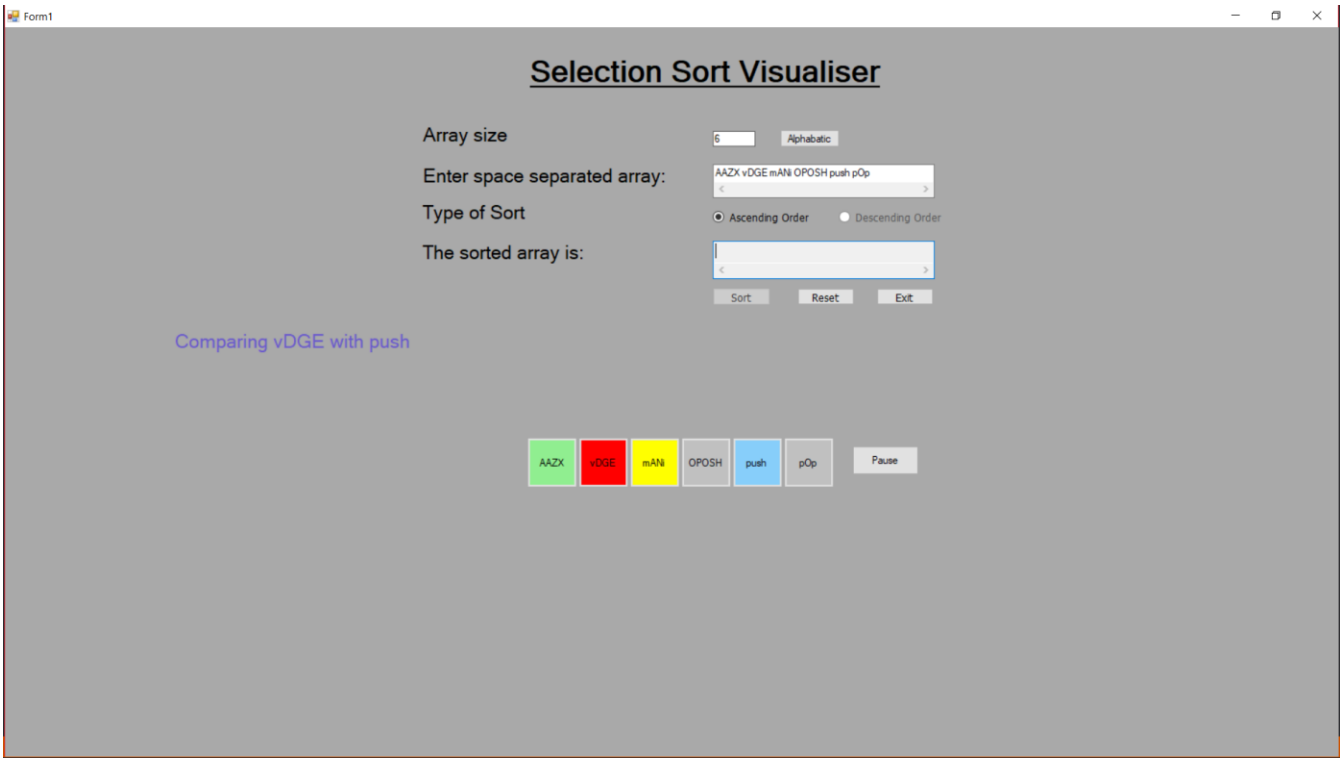
Hitting the **EXIT** command will lead to termination of the command.

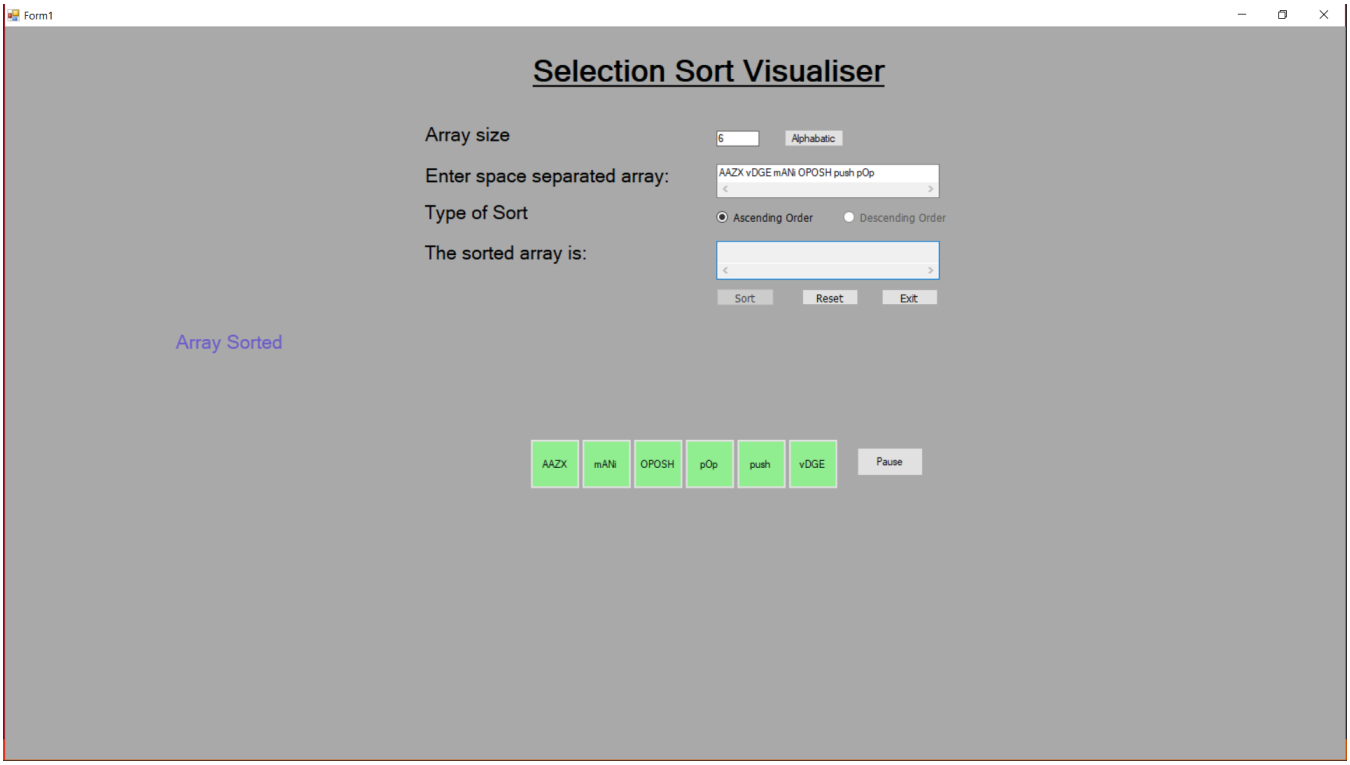
Some screen shots of working application:

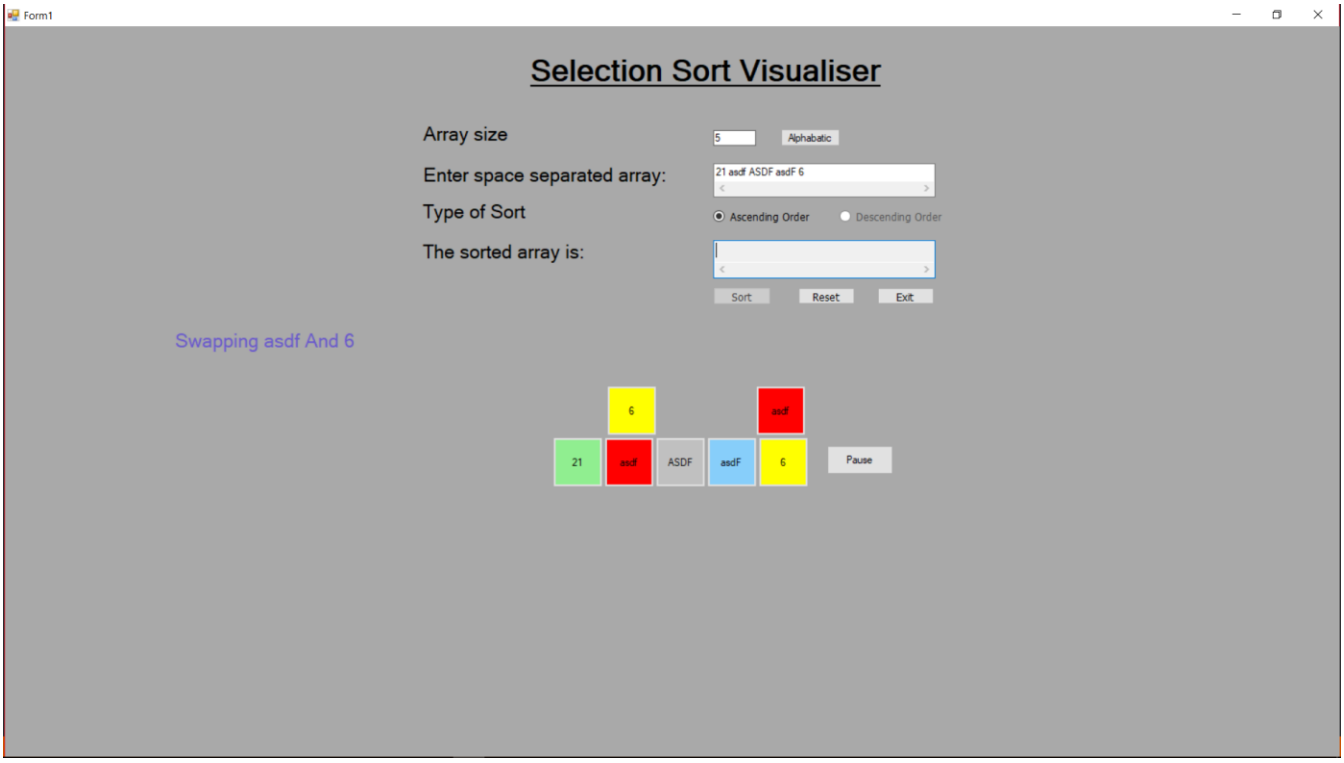


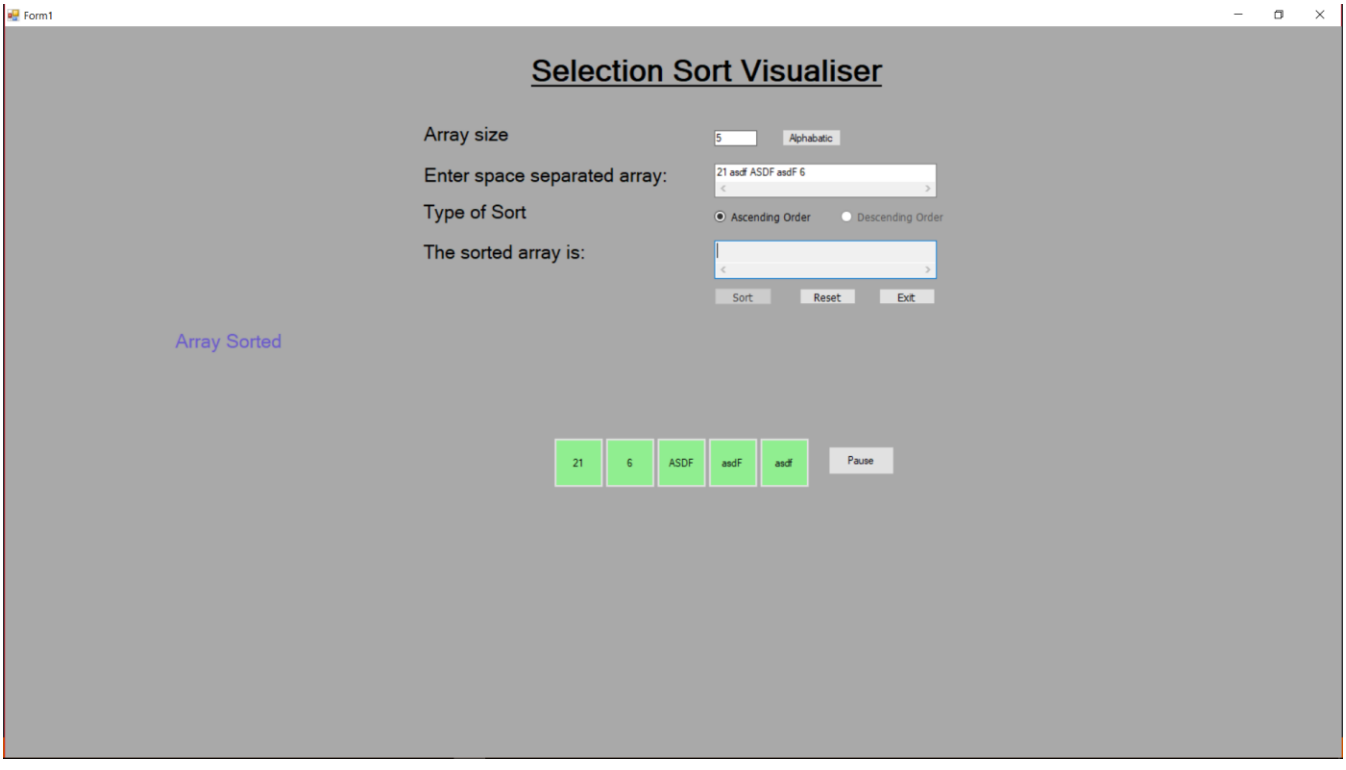

Form1
Selection Sort Visualiser
Array size
6
Numerical
Enter space separated array:
125 324 1 20 74 64
Type of Sort
Ascending Order
Descending Order
The sorted array is:
Sort
Reset
Exit
Swapping 324 And 20
20
324
1
324
125
20
74
64
Pause
Graph
350
300
250
200
150
100
50
0
-1
0
1
2
3
4
5
6


Form1
Selection Sort Visualiser
Array size
6
Numerical
Enter space separated array:
125 324 1 20 74 64
Type of Sort
Ascending Order
Descending Order
The sorted array is:
Sort
Reset
Exit
Swapping 324 And 74
74
324
1
20
64
324
74
125
Pause
Graph
350
300
250
200
150
100
50
0
-1
0
1
2
3
4
5
6

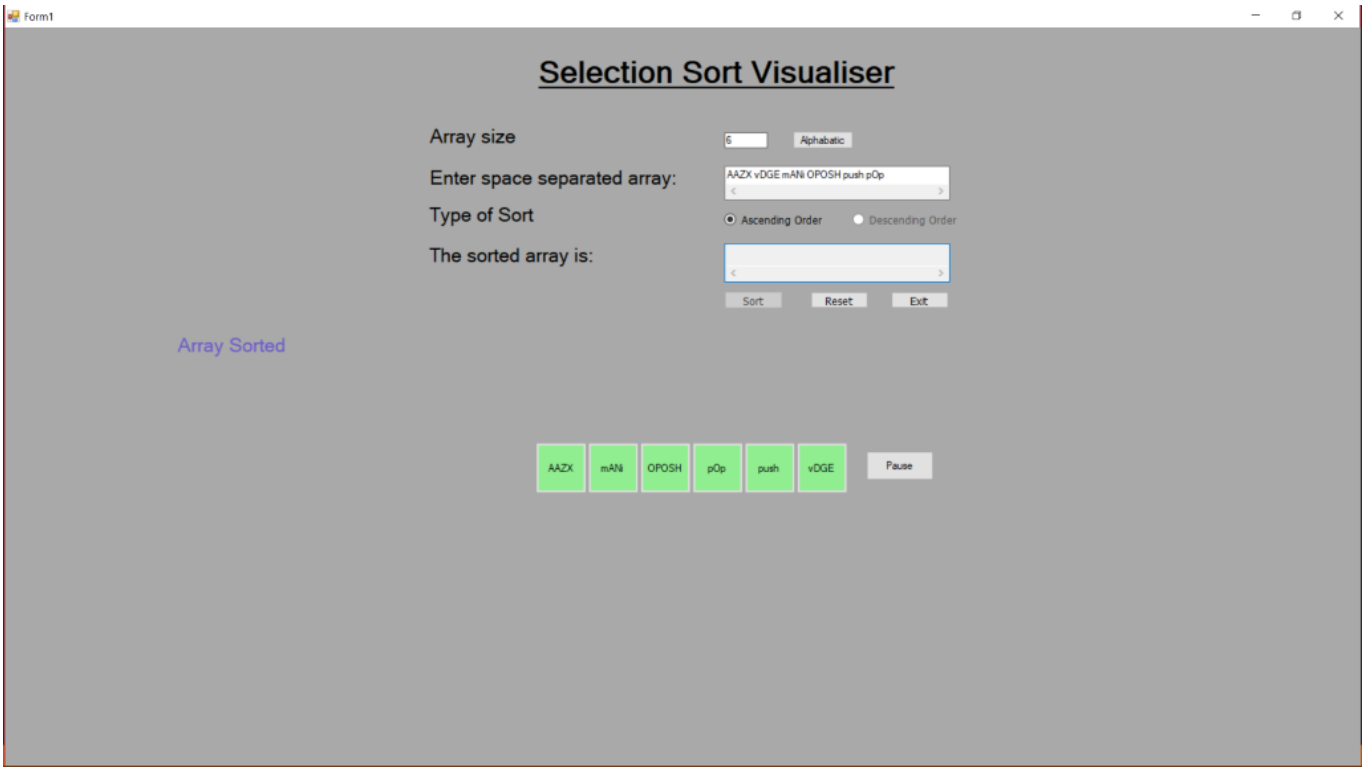

Form1
Selection Sort Visualiser
Array size
6
Alphabatic
Enter space separated array:
AAZX vDGE mANi OPOSH push pOp
Type of Sort
Ascending Order
Descending Order
The sorted array is:
Sort
Reset
Exit
Comparing vDGE with push
AAZX
vDGE
mANi
OPOSH
push
pOp
Pause


Form1
Selection Sort Visualiser
Array size
6
Alphabatic
Enter space separated array:
AAZX vDGE mANi OPOSH push pOp
Type of Sort
Ascending Order
Descending Order
The sorted array is:
Sort
Reset
Exit
Array Sorted
AAZX
mANi
OPOSH
pOp
push
vDGE
Pause



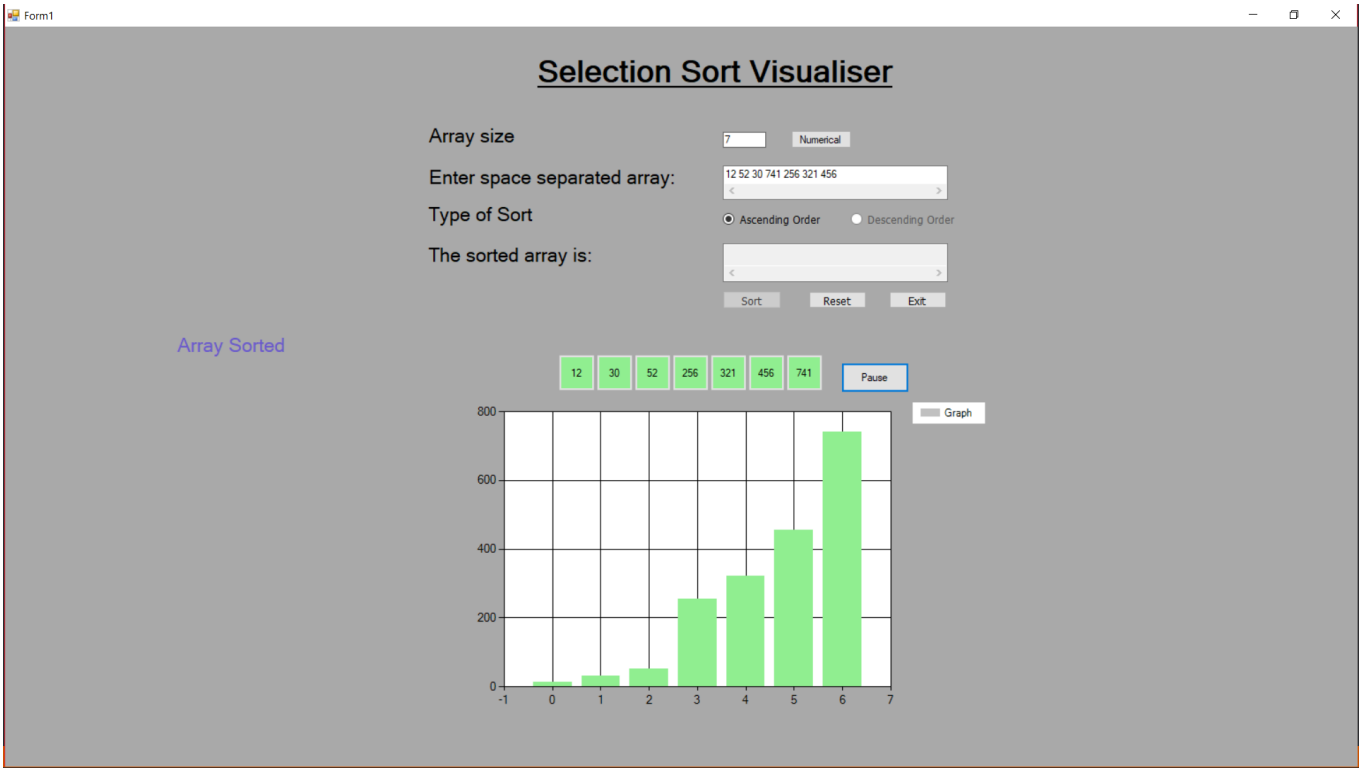


Some Inputs and Outputs:



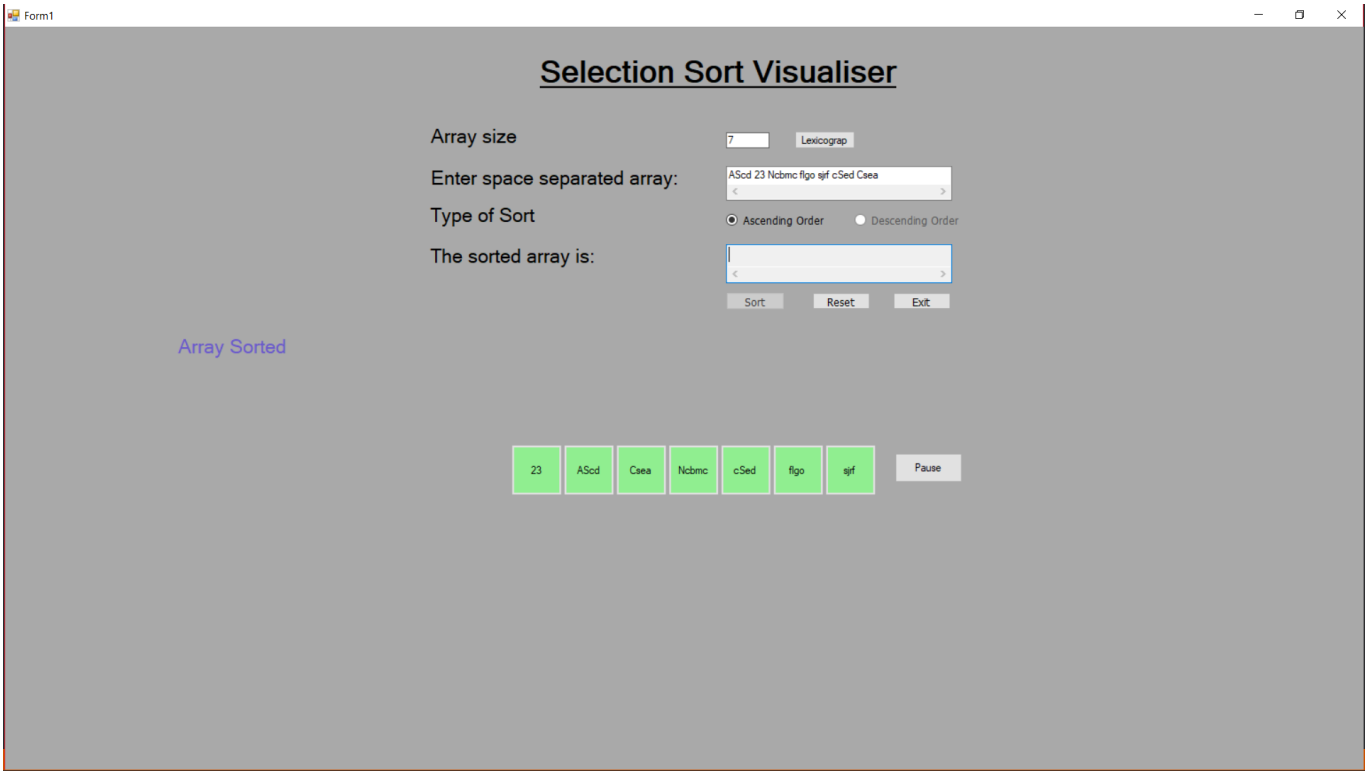
Input: AAZX vDGE mANi OPOSH push POP

Output: AAZX mANi OPOSH pOp push vDGE



Input: 12 52 30 741 256 321 456

Output: 12 30 52 256 321 456 741



Input: AScd 23 Ncbmc figo sjif cSed Csea

Output: 23 AScd Csea Ncbmc cSed figo sjif

Visualization:

For better animation effects User should enter <10 Decimals as input.

TWO MODES OF VISUALIZATION:

1. ARRAY
2. GRAPH

Red node represents the current element.

Yellow node represents the node having minimum value in that specific iteration.

White nodes are the nodes containing values greater than the minimum value of element in one pass of array.

Swapping of red and yellow nodes of array is shown just below the animated array for better understanding.

A proper delay is given for comparison of the numbers for better visualization.

There's a PAUSE button in the program which stops the animation so that you can have better look at the algorithm. To resume the process, press the PLAY button.

There are three links in the program: -

1. One link redirect to other sorting algorithm.
2. One link redirects to selection sort algorithm. There contains pseudocode and complete code to the algorithm.
3. Third link links to this guide i.e. the user documentation