Q1 c) Implement Bubble Sort Algorithm. Analyze its time complexity.

TIME COMPLEXITY:

1. Best Case: O(n)
2. Worst Case: O(n ^ 2)
3. Average Case: O(n ^ 2)

SPACE COMPLEXITY: O(1)

```
#include <iostream>
#include <vector>
#include <random>
using namespace std;
void bubbleSort(vector<int> & A)
{
    int n = int(A.size());
    for(int i = 0; i < n; ++i)
        for(int j = 0; j < n - i - 1; ++j)
            if (A[j] > A[j + 1])
                swap(A[j], A[j + 1]);
}
int main()
{
    vector<int> A(100);
    for(int i = 0; i < 100; ++i)
        A[i] = rand() % 1000;
    cout << "INITIAL STATE OF ARRAY:\n";
    for(int i = 0; i < 100; ++i)
        cout << A[i] << " ";
    cout << endl << endl;
    bubbleSort(A);
    cout << "SORTED ARRAY:\n";
    for(int i = 0; i < 100; ++i)
        cout << A[i] << " ";
}
```

OUTPUT:

```
INITIAL STATE OF ARRAY:
807 249 73 658 930 272 544 878 923 709 440 165 492 42 987 503 327 729 840
 612 303 169 709 157 560 933 99 278 816 335 97 826 512 267 810 633 979 149
 579 821 967 672 393 336 485 745 228 91 194 357 1 153 708 944 668 490 124
 196 530 903 722 666 549 24 801 853 977 408 228 933 298 981 635 13 865 814
 63 536 425 669 115 94 629 501 517 195 105 404 451 298 188 123 505 882 752
 566 716 337 438 144

SORTED ARRAY:
1 13 24 42 63 73 91 94 97 99 105 115 123 124 144 149 153 157 165 169 188
 194 195 196 228 228 249 267 272 278 298 298 303 327 335 336 337 357 393
 404 408 425 438 440 451 485 490 492 501 503 505 512 517 530 536 544 549
 560 566 579 612 629 633 635 658 666 668 669 672 708 709 709 716 722 729
 745 752 801 807 810 814 816 821 826 840 853 865 878 882 903 923 930 933
 933 944 967 977 979 981 987
Program ended with exit code: 0
```