National University
of computer and emerging sciences

Name: Ahmed Shayan

Roll Number: 21F-9135

Class:BS-CS-3B

Question No 3

Write a code to take input in 2 dynamic arrays as long as the user doesn&#39;t enter sentinel value. After you have finished taking input for one dynamic array start for the second. Their size depends on after entering how many elements the user enters the sentinel value and their size may be different. Now completely swap the contents of the two dynamic arrays.

Code

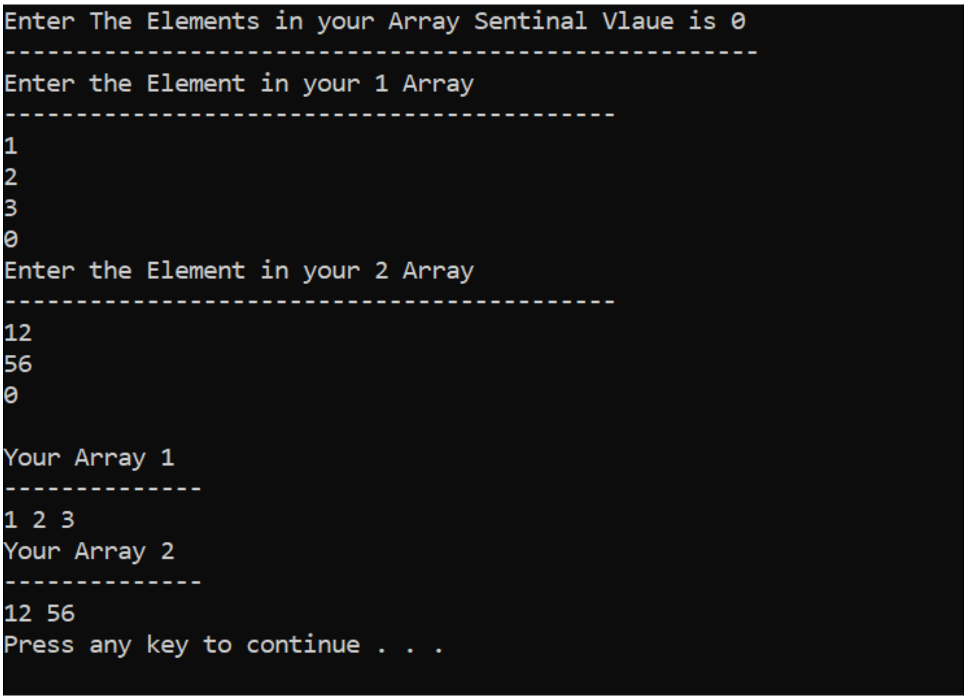
```
#include<iostream>
using namespace std;
int main() {
cout << "Enter The Elements in your Array Sentinal Vlaue is 0";
cout << endl;
cout << "---------------------------------------------------";
cout << endl;
int n = 1;
int* arr = NULL;
int* arrcp = NULL;
int* arr1 = NULL;
int* arrcp1 = NULL;
int size = 1;
int size1 = 1;
int j = 0;
arr = new int[size];
arr1 = new int[size];
```

```
for (int i = 0; i < 2; i++) {
cout << "Enter the Element in your " << i + 1 << " Array";
cout << endl;
cout << "-------------------------------------------";
cout << endl;
if (i == 0) {
while (n != 0) {
cin >> n;
if (n == 0) {
break;
}
else {
arr[j] = n;
size = size + 1;
arrcp = new int[size];
for (int k = 0; k < size - 1; k++) {
arrcp[k] = arr[k];
}
delete arr;
arr = new int[size];
for (int k = 0; k < size - 1; k++) {
arr[k] = arrcp[k];
}
delete arrcp;
}
j = j + 1;
}
}
n = 1;
j = 0;
if (i == 1) {
```

```
while (n != 0) {

cin >> n;

if (n == 0) {

break;

}

else {

arr1[j] = n;

size1 = size1 + 1;

arrcp1 = new int[size1];

for (int k = 0; k < size1 - 1; k++) {

arrcp1[k] = arr1[k];

}

delete arr1;

arr1 = new int[size1];

for (int k = 0; k < size1 - 1; k++) {

arr1[k] = arrcp1[k];

}

delete arrcp1;

}

j = j + 1;

}

}

}

cout << endl;

for (int k = 0; k < 2; k++) {

if (k == 0) {

cout << "Your Array 1";

cout << endl;

cout << "--------------";

cout << endl;

for (int i = 0; i < size - 1; i++) {
```

```
cout << arr[i]<<" ";
}
}
if (k == 1) {
cout << endl;
cout << "Your Array 2";
cout << endl;
cout << "--------------";
cout << endl;
for (int i = 0; i < size1 - 1; i++) {
cout << arr1[i]<<" ";
}
}
}
cout << endl;
system("pause");
return 0;
}
```

OUTPUT



```
Enter The Elements in your Array Sentinal Vlaue is 0
-----------------------------------------------------
Enter the Element in your 1 Array
-------------------------------------------
1
2
3
0
Enter the Element in your 2 Array
-------------------------------------------
12
56
0

Your Array 1
--------------
1 2 3
Your Array 2
--------------
12 56
Press any key to continue . . .
```

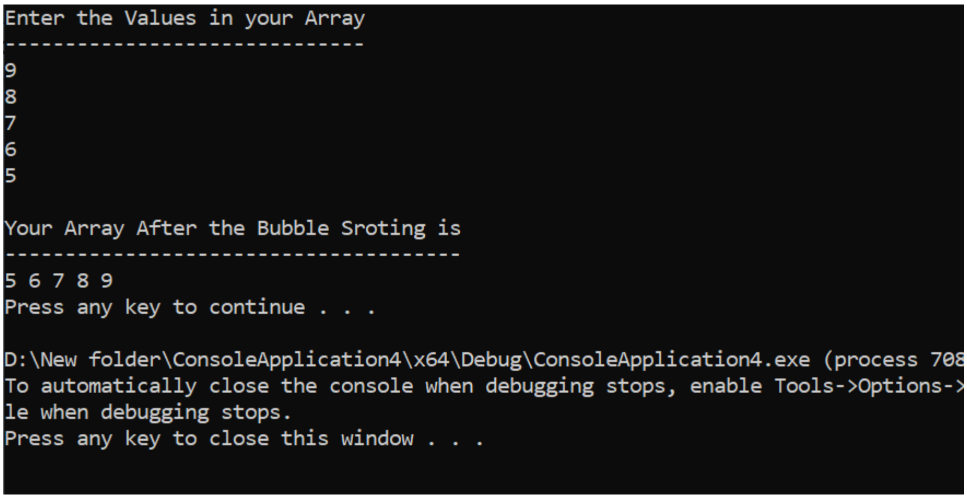
Question no 4

Implement another function; pass the array (again using a pointer) to it. The function should then sort the array using the Bubble sort. The function should use pointers for all computations (counter variables, traversal, swap).

Code

```
#include<iostream>
using namespace std;
void Buble_Sorting(int* a) {
cout << "Enter the Values in your Array";
cout << endl;
cout << "------------------------------";
cout << endl;
int* i = new int;
int* j = new int;
for (*i = 0; *i < 5; *i=*i+1) {
cin >> *(a + *i);
}
for (*i = 0; *i < 5; *i = *i + 1) {
for (*j = *i + 1; *j < 5; *j=*j+1) {
if (*(a + *i) > *(a + *j)) {
swap(*(a + *i), *(a + *j));
}
}
}
cout << endl;
cout << "Your Array After the Bubble Sroting is";
cout << endl;
cout << "-------------------------------------";
cout << endl;
for (*i = 0; *i < 5; *i=*i+1) {
cout << *(a + *i) << " ";
}
}
int main() {
int a[5];
Buble_Sorting(a);
cout << endl;
system("pause");
return 0;
}
```

<u>OUTPUT</u>



```
Enter the Values in your Array
------------------------------
9
8
7
6
5

Your Array After the Bubble Sroting is
--------------------------------------
5 6 7 8 9
Press any key to continue . . .

D:\New folder\ConsoleApplication4\x64\Debug\ConsoleApplication4.exe (process 708
To automatically close the console when debugging stops, enable Tools->Options->
le when debugging stops.
Press any key to close this window . . .
```