Zana Osman

September 19th, 2023

200489300

1) We cannot, this is due to the size of the array not being the same. The array of 50 cannot be assigned to the array of 30 as they are not the same size. If you were to try and do this it would result in an error as the first 30 numbers would copy, however the last 20 would not and will result in garbage value.

2) Using Address = B + W (i * n + j)

    nArray[11][19] and nArray[20][30]

    B = Base = 1650

    W = Weight (Storage size in bytes) = 4

    i = Row Index = 11

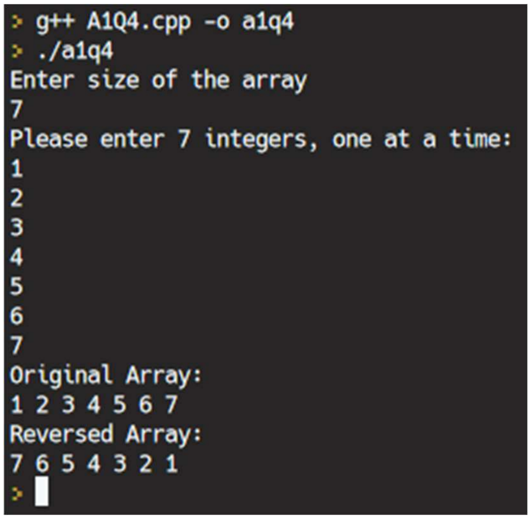
    n = Number of Columns = 30 + 1

    j = Column Index = 19

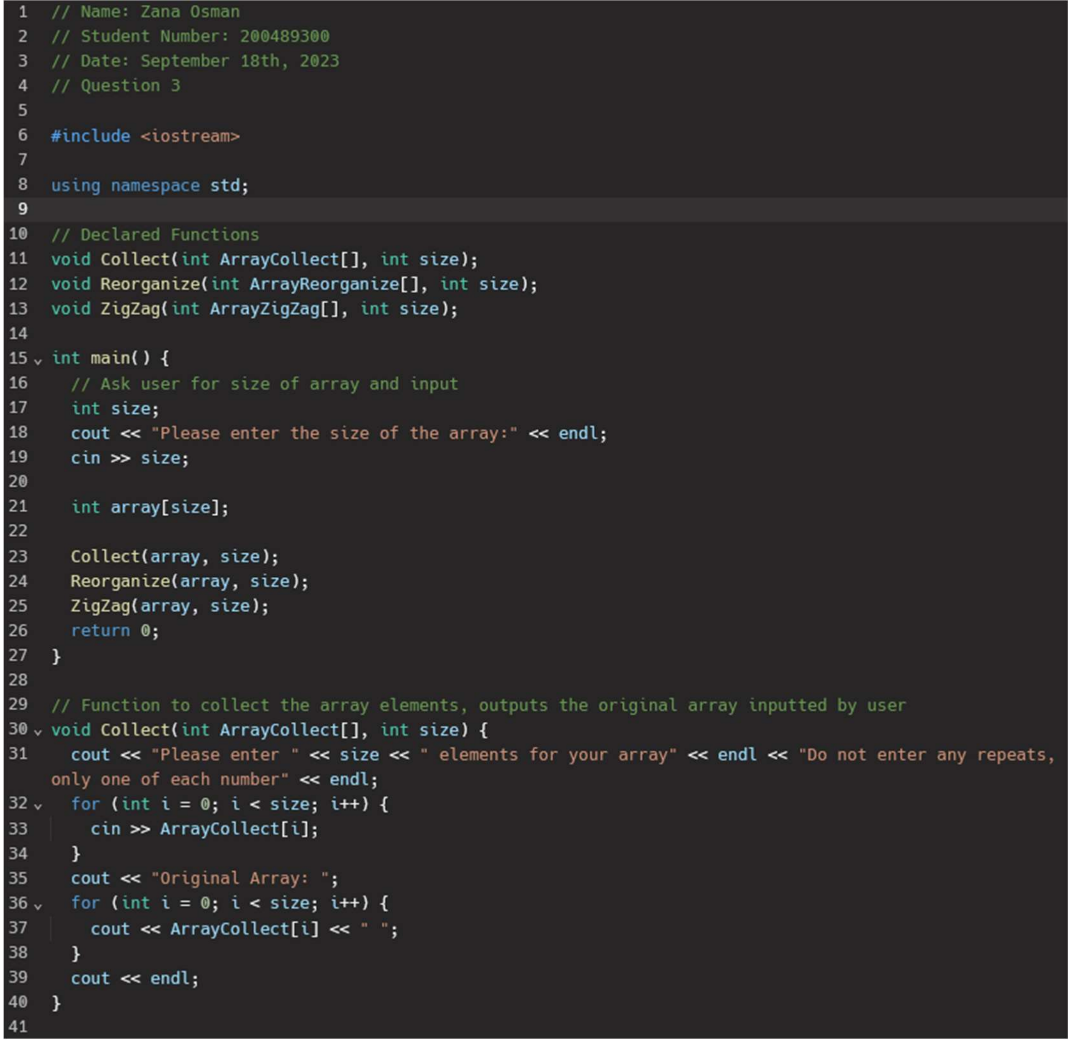
   Address = 1650 + 4 * (11 * 31 + 19)

   Address = 3090

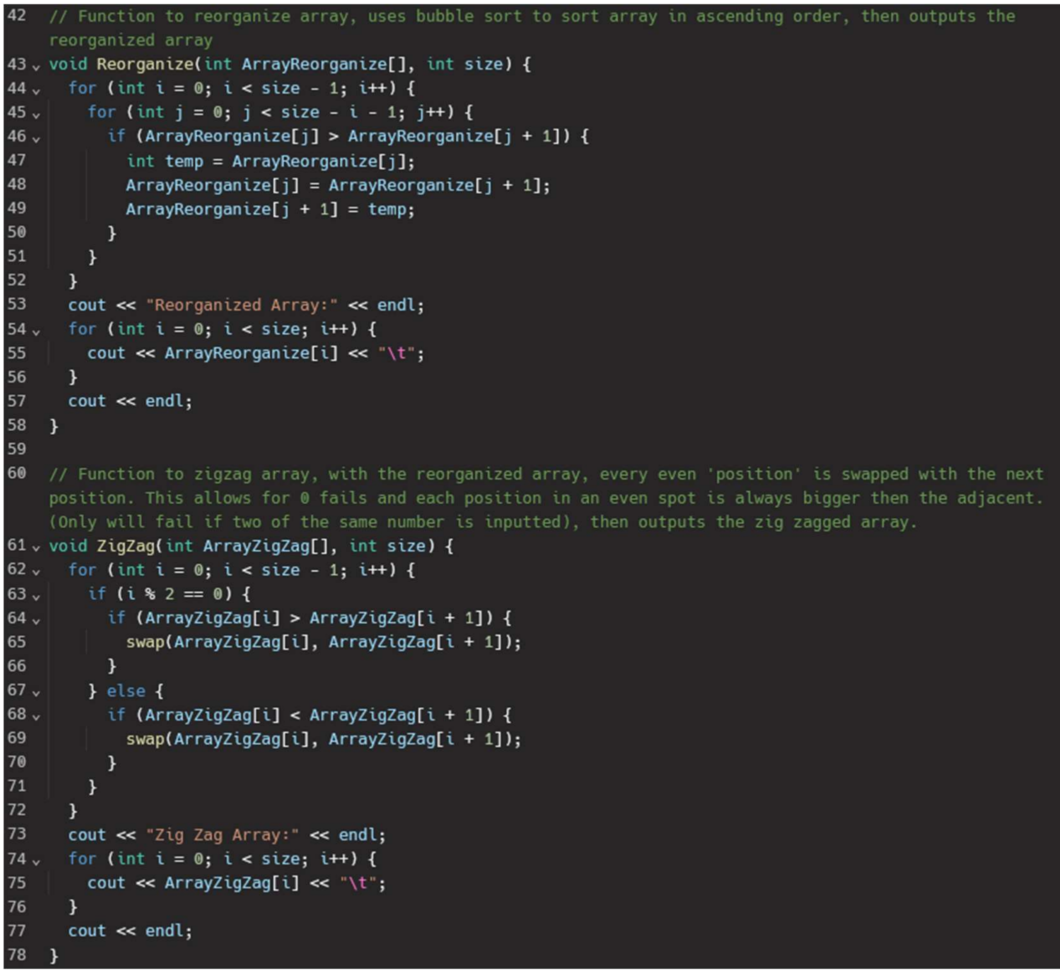
Question 3, photos with code



```
> g++ A1Q4.cpp -o a1q4
> ./a1q4
Enter size of the array
7
Please enter 7 integers, one at a time:
1
2
3
4
5
6
7
Original Array:
1 2 3 4 5 6 7
Reversed Array:
7 6 5 4 3 2 1
>
```

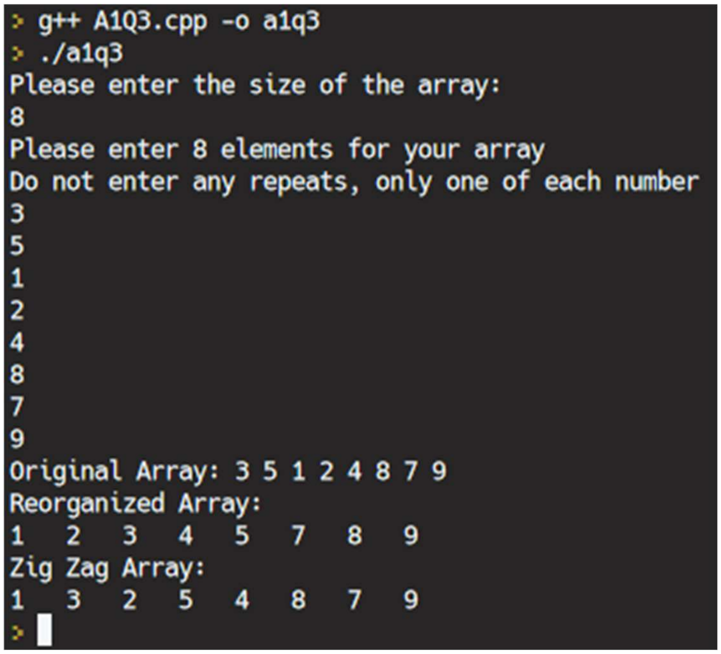


```cpp
// Name: Zana Osman
// Student Number: 200489300
// Date: September 18th, 2023
// Question 3

#include <iostream>

using namespace std;

// Declared Functions
void Collect(int ArrayCollect[], int size);
void Reorganize(int ArrayReorganize[], int size);
void ZigZag(int ArrayZigZag[], int size);

int main() {
  // Ask user for size of array and input
  int size;
  cout << "Please enter the size of the array:" << endl;
  cin >> size;

  int array[size];

  Collect(array, size);
  Reorganize(array, size);
  ZigZag(array, size);
  return 0;
}

// Function to collect the array elements, outputs the original array inputted by user
void Collect(int ArrayCollect[], int size) {
  cout << "Please enter " << size << " elements for your array" << endl << "Do not enter any repeats, only one of each number" << endl;
  for (int i = 0; i < size; i++) {
    cin >> ArrayCollect[i];
  }
  cout << "Original Array: ";
  for (int i = 0; i < size; i++) {
    cout << ArrayCollect[i] << " ";
  }
  cout << endl;
}

```

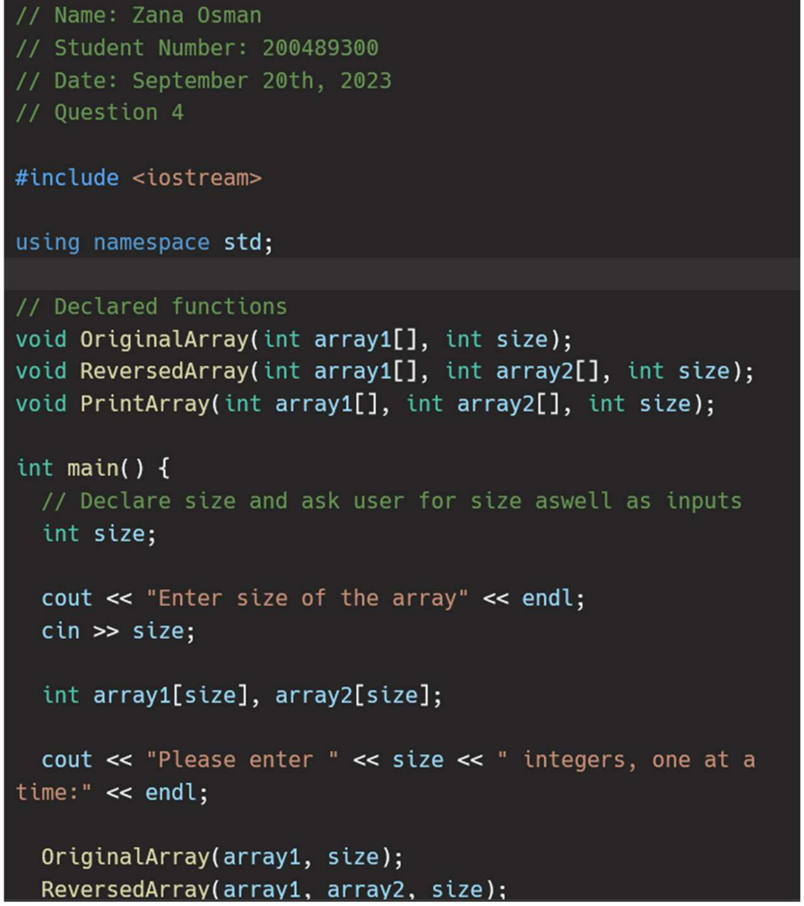


```cpp
// Function to reorganize array, uses bubble sort to sort array in ascending order, then outputs the reorganized array
void Reorganize(int ArrayReorganize[], int size) {
  for (int i = 0; i < size - 1; i++) {
    for (int j = 0; j < size - i - 1; j++) {
      if (ArrayReorganize[j] > ArrayReorganize[j + 1]) {
        int temp = ArrayReorganize[j];
        ArrayReorganize[j] = ArrayReorganize[j + 1];
        ArrayReorganize[j + 1] = temp;
      }
    }
  }
  cout << "Reorganized Array:" << endl;
  for (int i = 0; i < size; i++) {
    cout << ArrayReorganize[i] << "\t";
  }
  cout << endl;
}

// Function to zigzag array, with the reorganized array, every even 'position' is swapped with the next position. This allows for 0 fails and each position in an even spot is always bigger then the adjacent. (Only will fail if two of the same number is inputted), then outputs the zig zagged array.
void ZigZag(int ArrayZigZag[], int size) {
  for (int i = 0; i < size - 1; i++) {
    if (i % 2 == 0) {
      if (ArrayZigZag[i] > ArrayZigZag[i + 1]) {
        swap(ArrayZigZag[i], ArrayZigZag[i + 1]);
      }
    } else {
      if (ArrayZigZag[i] < ArrayZigZag[i + 1]) {
        swap(ArrayZigZag[i], ArrayZigZag[i + 1]);
      }
    }
  }
  cout << "Zig Zag Array:" << endl;
  for (int i = 0; i < size; i++) {
    cout << ArrayZigZag[i] << "\t";
  }
  cout << endl;
}
```

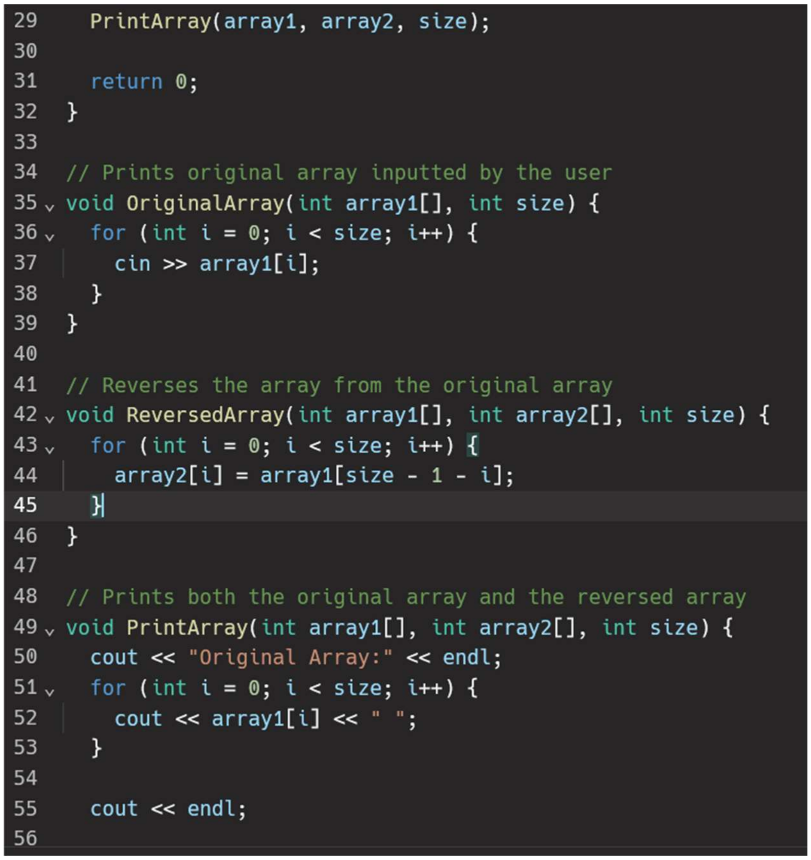
Question 4, photos of code



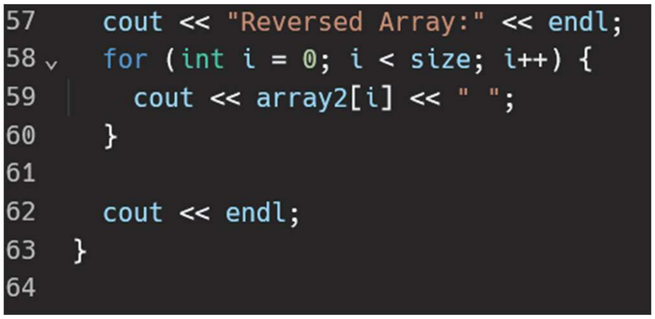
```
> g++ A1Q3.cpp -o a1q3
> ./a1q3
Please enter the size of the array:
8
Please enter 8 elements for your array
Do not enter any repeats, only one of each number
3
5
1
2
4
8
7
9
Original Array: 3 5 1 2 4 8 7 9
Reorganized Array:
1   2   3   4   5   7   8   9
Zig Zag Array:
1   3   2   5   4   8   7   9
>
```



```cpp
// Name: Zana Osman
// Student Number: 200489300
// Date: September 20th, 2023
// Question 4

#include <iostream>

using namespace std;

// Declared functions
void OriginalArray(int array1[], int size);
void ReversedArray(int array1[], int array2[], int size);
void PrintArray(int array1[], int array2[], int size);

int main() {
  // Declare size and ask user for size aswell as inputs
  int size;

  cout << "Enter size of the array" << endl;
  cin >> size;

  int array1[size], array2[size];

  cout << "Please enter " << size << " integers, one at a
time:" << endl;

  OriginalArray(array1, size);
  ReversedArray(array1, array2, size);
```



```cpp
  PrintArray(array1, array2, size);

  return 0;
}

// Prints original array inputted by the user
void OriginalArray(int array1[], int size) {
  for (int i = 0; i < size; i++) {
    cin >> array1[i];
  }
}

// Reverses the array from the original array
void ReversedArray(int array1[], int array2[], int size) {
  for (int i = 0; i < size; i++) {
    array2[i] = array1[size - 1 - i];
  }
}

// Prints both the original array and the reversed array
void PrintArray(int array1[], int array2[], int size) {
  cout << "Original Array:" << endl;
  for (int i = 0; i < size; i++) {
    cout << array1[i] << " ";
  }

  cout << endl;

```



```cpp
  cout << "Reversed Array:" << endl;
  for (int i = 0; i < size; i++) {
    cout << array2[i] << " ";
  }

  cout << endl;
}

```