Rayan Ahmed

23i-0018

Q1:

```cpp
#include <iostream>

using namespace std;

void printArray(int *arr, int size)
{
    for (int i = 0; i < size; i++)
        cout << *(arr + i) << " ";
    cout << endl;
}

void appendValue(int *&arr, int size, int num)
{
    int *temp = new int[size];

    for (int i = 0; i < size; i++)
        *(temp + i) = *(arr + i);

    delete[] arr;
    arr = new int[size + 1];

    for (int i = 0; i < size + 1; i++)
        *(arr + i) = *(temp + i);

    *(arr + size) = num;

    delete[] temp;
}

void swapArrays(int *&arr1, int &size1, int *&arr2, int &size2)
{
    int *temp = new int[size1];

    for (int i = 0; i < size1; i++)
        *(temp + i) = *(arr1 + i);

    delete[] arr1;
    arr1 = new int[size2];
```

```
    for (int i = 0; i < size2; i++)
        *(arr1 + i) = *(arr2 + i);

    delete[] arr2;
    arr2 = new int[size1];

    for (int i = 0; i < size1; i++)
        *(arr2 + i) = *(temp + i);

    size1 = size1 + size2;
    size2 = size1 - size2;
    size1 = size1 - size2;

    delete[] temp;
}

int main()
{
    int *arr1 = nullptr, *arr2 = nullptr;
    int size1 = 0, size2 = 0;

    cout << "Array 1: " << endl;
    for (size1; true; size1++)
    {
        int num;

        cout << "Enter a number(Enter a negative value to quit): ";
        cin >> num;

        if (num < 0)
            break;

        appendValue(arr1, size1, num);
    }

    cout << "Array 2: " << endl;
    for (size2; true; size2++)
    {
        int num;

        cout << "Enter a number(Enter a negative value to quit): ";
        cin >> num;

        if (num < 0)
            break;

        appendValue(arr2, size2, num);
```

```
    }

    cout << "Original Arrays: " << endl;
    cout << "Array 1: ";
    printArray(arr1, size1);
    cout << "Array 2: ";
    printArray(arr2, size2);

    swapArrays(arr1, size1, arr2, size2);
    cout << "Swapped Arrays: " << endl;
    cout << "Array 1: ";
    printArray(arr1, size1);
    cout << "Array 2: ";
    printArray(arr2, size2);

    delete[] arr1;
    delete[] arr2;

    return 0;
}
```

Output:

```
Array 1:
Enter a number(Enter a negative value to quit): 1
Enter a number(Enter a negative value to quit): 3
Enter a number(Enter a negative value to quit): 5
Enter a number(Enter a negative value to quit): 7
Enter a number(Enter a negative value to quit): 9
Enter a number(Enter a negative value to quit): -1
Array 2:
Enter a number(Enter a negative value to quit): 2
Enter a number(Enter a negative value to quit): 4
Enter a number(Enter a negative value to quit): 6
Enter a number(Enter a negative value to quit): 8
Enter a number(Enter a negative value to quit): -1
Original Arrays:
Array 1: 1 3 5 7 9
Array 2: 2 4 6 8
Swapped Arrays:
Array 1: 2 4 6 8
Array 2: 1 3 5 7 9
```

Q2:

```
#include <iostream>
#include <ctime>

using namespace std;

void displayMatrix(int **arr)
{
    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 3; j++)
            cout << *(*(arr + i) + j) << " ";
        cout << endl;
```

```
    }
    cout << endl;
}

void clearMatrix(int **&arr)
{
    for (int i = 0; i < 3; i++)
        for(int j = 0; j < 3; j++)
            *(*(arr + i) + j) = 0;
}

void addMatrix(int **arr1, int **arr2, int **&result)
{
    for (int i = 0; i < 3; i++)
        for (int j = 0; j < 3; j++)
            *(*(result + i) + j) = *(*(arr1 + i) + j) + *(*(arr2 + i) +
j);
}

void multiplyMatrix(int **arr1, int **arr2, int **&result)
{
    for (int i = 0; i < 3; i++)
        for (int j = 0; j < 3; j++)
            for (int k = 0; k < 3; k++)
                *(*(result + i) + j) += *(*(arr1 + i) + k) * *(*(arr2 +
k) + j);
}

int main()
{
    srand(time(nullptr));

    int **arr1 = new int *[3];
    for (int i = 0; i < 3; i++)
        *(arr1 + i) = new int[3];

    for (int i = 0; i < 3; i++)
        for (int j = 0; j < 3; j++)
            *(*(arr1 + i) + j) = rand() % 20;

    int **arr2 = new int *[3];
    for (int i = 0; i < 3; i++)
        *(arr2 + i) = new int[3];

    for (int i = 0; i < 3; i++)
        for (int j = 0; j < 3; j++)
            *(*(arr2 + i) + j) = rand() % 20;
```

```
    cout << "Array 1: " << endl;
    displayMatrix(arr1);

    cout << "Array 2: " << endl;
    displayMatrix(arr2);

    int **result = new int *[3];
    for (int i = 0; i < 3; i++)
        *(result + i) = new int[3];

    addMatrix(arr1, arr2, result);
    cout << "Addition: " << endl;
    displayMatrix(result);

    clearMatrix(result);

    multiplyMatrix(arr1, arr2, result);
    cout << "Multiplication: " << endl;
    displayMatrix(result);

    return 0;
}
```

Output:

```
Array 1:
8 17 8
2 13 19
3 19 16

Array 2:
11 5 13
9 10 5
1 15 5

Addition:
19 22 21
11 23 24
4 34 21

Multiplication:
249 330 229
158 425 186
220 445 214
```

Q3:

```
#include <iostream>
#include <ctime>

using namespace std;

void printArray(int *arr, int size)
{
    for (int i = 0; i < size; i++)
        cout << *(arr + i) << " ";
```

```
    cout << endl;
}

int findMax(int *arr, int size)
{
    int max = *arr;

    for (int i = 0; i < size; i++)
        if (*(arr + i) > max)
            max = *(arr + i);

    return max;
}

int findMin(int *arr, int size)
{
    int min = *arr;

    for (int i = 0; i < size; i++)
        if (*(arr + i) < min)
            min = *(arr + i);

    return min;
}

int main()
{
    srand(time(nullptr));
    int size = rand() % 10 + 3;

    int *arr = new int[size];
    for (int i = 0; i < size; i++)
        *(arr + i) = rand() % 50;

    cout << "Array: ";
    printArray(arr, size);

    cout << "Largest Element: " << findMax(arr, size) << endl;
    cout << "Smallest Element: " << findMin(arr, size) << endl;

    return 0;
}
```

Output:

```
Array: 40 49 13 40 18 37
Largest Element: 49
Smallest Element: 13
```

Q4:

```cpp
#include <iostream>
#include <ctime>

using namespace std;

void printArray(int *arr, int size)
{
    for (int i = 0; i < size; i++)
        cout << *(arr + i) << " ";
    cout << endl;
}

void bubbleSort(int *&arr, int size)
{
    for (int i = 0; i < size; i++)
        for (int j = 0; j < size - i - 1; j++)
            if (*(arr + j) > *(arr + j + 1))
            {
                *(arr + j) = *(arr + j) + *(arr + j + 1);
                *(arr + j + 1) = *(arr + j) - *(arr + j + 1);
                *(arr + j) = *(arr + j) - *(arr + j + 1);
            }
}

int main()
{
    srand(time(nullptr));
    int size = rand() % 10 + 3;

    int *arr = new int[size];
    for (int i = 0; i < size; i++)
        *(arr + i) = rand() % 50;

    cout << "Original Array: ";
    printArray(arr, size);

    bubbleSort(arr, size);

    cout << "Sorted Array: ";
    printArray(arr, size);

    return 0;
}
```

Output:

```
Original Array: 9 9 13 49 28 19 32 29 15 9 5
Sorted Array: 5 9 9 9 13 15 19 28 29 32 49
```

Q5:

```cpp
#include <iostream>
#include <ctime>

using namespace std;

void printArray(int *arr, int size)
{
    for (int i = 0; i < size; i++)
        cout << *(arr + i) << " ";
    cout << endl;
}

void rotateRight(int *&arr, int size, int steps)
{
    int *temp = new int[size];

    for (int i = 0, j = steps; i < size; i++, j++)
        *(temp + j % size) = *(arr + i);

    for (int i = 0; i < size; i++)
        *(arr + i) = *(temp + i);
}

int main()
{
    srand(time(nullptr));
    int size = rand() % 10 + 3;

    int *arr = new int[size];
    for (int i = 0; i < size; i++)
        *(arr + i) = rand() % 50;

    cout << "Original Array: ";
    printArray(arr, size);

    int steps;
    cout << "Enter steps: ";
    cin >> steps;

    rotateRight(arr, size, steps);
    cout << "Rotated Array: ";
```

```
    printArray(arr, size);

    return 0;
}
```

Output:

```
Original Array: 38 6 15 38 4 38 0
Enter steps: 3
Rotated Array: 4 38 0 38 6 15 38
```