



```

//Prime Number Checker & Factor Giver
#include <iostream>
#include <cmath>
using namespace std;

// Function to check if a number is prime
int PrimeChecker(int n) {
    if (n <= 1) return 0;
    for (int i = 2; i <= sqrt(n); ++i) {
        if (n % i == 0) return 0;
    }
    return 1;
}

// Function to find the next prime greater than n
int NextPrime(int n) {
    int next = n + 1;
    while (true) {
        if (PrimeChecker(next) == 1) return next;
        ++next;
    }
}

//Finding factors if not prime
void FactorFinder(int n) {
    cout << "Factors of " << n << ": ";
    for (int i = 1; i <= n / 2; ++i) {
        if (n % i == 0) {
            cout << i << " ";
        }
    }
    cout << n << endl;
}

int main() {
    int n;
    cout << "Enter a positive integer: ";
    cin >> n;

    if (PrimeChecker(n)) {
        cout << n << " is a prime number." << endl;
        cout << "The next prime number is " << NextPrime(n) << "." << endl;
    } else {
        cout << n << " is not a prime number." << endl;
        FactorFinder(n);
    }

    return 0;
}

```

```

#include <iostream>
#include <climits>
using namespace std;

// Function to reverse the array
void ArrayReverser(int arr[], int size) {
    int start = 0;
    int end = size - 1;
    while (start < end) {
        int temp = arr[start];
        arr[start] = arr[end];
        arr[end] = temp;

        start++;
        end--;
    }
}

void printArray(int arr[], int size) {
    for (int i = 0; i < size; i++) {
        cout << arr[i];
        if (i < size - 1) {
            cout << ", ";
        }
    }
    cout << endl;
}

void FindingNumbers(int arr[], int size) {
    if (size < 2) {
        cout << "Array should have at least 2 elements" << endl;
        return;
    }

    int largest = arr[0], smallest = arr[0];
    int secondLargest = INT_MIN, secondSmallest = INT_MAX;
    for (int i = 1; i < size; i++) {
        if (arr[i] > largest) {
            secondLargest = largest;
            largest = arr[i];
        } else if (arr[i] > secondLargest && arr[i] != largest) {
            secondLargest = arr[i];
        }

        if (arr[i] < smallest) {
            secondSmallest = smallest;
            smallest = arr[i];
        } else if (arr[i] < secondSmallest && arr[i] != smallest) {
            secondSmallest = arr[i];
        }
    }

    if (secondLargest == INT_MIN) {
        cout << "Second Largest element does not exist." << endl;
    } else {
        cout << "Second Largest element is: " << secondLargest << endl;
    }

    if (secondSmallest == INT_MAX) {
        cout << "Second Smallest element does not exist." << endl;
    } else {
        cout << "Second Smallest element is: " << secondSmallest << endl;
    }
}

int main() {
    int size;

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

    if (size <= 0) {
        cout << "Enter valid size (greater than 0)" << endl;
        return 1;
    }

    int* arr = new int[size];

    cout << "Enter " << size << " elements:" << endl;
    for (int i = 0; i < size; i++) {
        cout << "Element " << i + 1 << ": ";
        cin >> arr[i];
    }

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

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

    FindingNumbers(arr, size);

    // Free the dynamically allocated memory
    delete[] arr;

    return 0;
}

```

```

#include <iostream>
#include <string>
#include <cctype>
using namespace std;

string lowercase_string(string str) {
    string result = "";
    for(char c : str) {
        if(c != ' ') {
            result += tolower(c);
        }
    }
    return result;
}

bool palindrome_checker(string str) {
    string lowered_string = lowercase_string(str);
    int start = 0;
    int end = lowered_string.length() - 1;

    while(start < end) {
        if(lowered_string[start] != lowered_string[end]) {
            return false;
        }
        start++;
        end--;
    }
    return true;
}

void frequency_counter(string str) {
    int frequency[26] = {0};
    for(char c : str) {
        if(isalpha(c)) {
            frequency[tolower(c) - 'a']++;
        }
    }

    cout << "Character frequencies:" << endl;
    for(int i = 0; i < 26; i++) {
        if(frequency[i] > 0) {
            cout << (char)(i + 'a') << ": " << frequency[i] << endl;
        }
    }
}

string replacer(string str, char replacement) {
    string result = str;
    for(int i = 0; i < result.length(); i++) {
        char c = tolower(result[i]);
        if(c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u') {
            result[i] = replacement;
        }
    }
    return result;
}

int main() {
    string input;

    cout << "Enter a string: ";
    getline(cin, input);

    if(palindrome_checker(input)) {
        cout << "\nThe string IS a palindrome!" << endl;
    } else {
        cout << "\nThe string is NOT a palindrome." << endl;
    }

    cout << "\n";
    frequency_counter(input);

    char replacement;
    cout << "Enter a character to replace vowels with: ";
    cin >> replacement;

    string modifiedString = replacer(input, replacement);
    cout << "\nString with vowels replaced: " << modifiedString <<
endl;
    return 0;
}

```



```
● ● ●

#include <iostream>
using namespace std;

void printSpiralPattern(int n) {
    int spiral[n][n];
    int startRow = 0, endRow = n - 1;
    int startCol = 0, endCol = n - 1;
    int value = 1;

    while (startRow <= endRow && startCol <= endCol) {
        for (int i = startCol; i <= endCol; i++) {
            spiral[startRow][i] = value++;
        }
        startRow++;

        for (int i = startRow; i <= endRow; i++) {
            spiral[i][endCol] = value++;
        }
        endCol--;

        if (startRow <= endRow) {
            for (int i = endCol; i >= startCol; i--) {
                spiral[endRow][i] = value++;
            }
            endRow--;
        }

        if (startCol <= endCol) {
            for (int i = endRow; i >= startRow; i--) {
                spiral[i][startCol] = value++;
            }
            startCol++;
        }
    }

    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            cout << spiral[i][j] << " ";
        }
        cout << endl;
    }
}

int main() {
    int n;
    cout << "Enter the size of the spiral matrix (n): ";
    cin >> n;

    printSpiralPattern(n);

    return 0;
}
```

```

#include <iostream>
using namespace std;

void rotateMatrix90DegreesClockwise(int matrix[][3], int n) {
    for (int layer = 0; layer < n / 2; layer++) {
        int first = layer;
        int last = n - layer - 1;
        for (int i = first; i < last; i++) {
            int offset = i - first;
            int top = matrix[first][i];

            matrix[first][i] = matrix[last - offset][first];
            matrix[last - offset][first] = matrix[last][last - offset];
            matrix[last][last - offset] = matrix[i][last];
            matrix[i][last] = top;
        }
    }
}

void printMatrix(int matrix[][3], int n) {
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            cout << matrix[i][j] << " ";
        }
        cout << endl;
    }
}

int main() {
    int n = 3;
    int matrix[3][3] = {
        {1, 2, 3},
        {4, 5, 6},
        {7, 8, 9}
    };

    cout << "Original Matrix:" << endl;
    printMatrix(matrix, n);

    rotateMatrix90DegreesClockwise(matrix, n);

    cout << "\nMatrix after 90 degrees clockwise rotation:" << endl;
    printMatrix(matrix, n);

    return 0;
}

```