**Assignment-1**

**// Q1. Number Manipulation and Prime Numbers**

#include<iostream>

using namespace std;

int prime(int num){

    if (num<1)

        return 0;

    for(int i=2;i<num;i++){

        if (num % i == 0){

            return 0;

        }

    }

    return 1;

}

int nextPrime(int num){

    int next = num + 1;

    while (!prime(next)) {

        next++;

    }

    return next;

}

void factors(int num){

    for(int i=1;i<=num;i++){

        if (num % i == 0){

            cout<<i<<" ";

        }

    }

    cout<<endl;

}

int main(){

    int n;

    cout<<"Enter any positive integer: ";

    cin>>n;

    if (prime(n)){

        cout<<"The given number is prime."<<endl;

        cout<<"The next prime number greater than the given number is: "<<nextPrime(n);

    }

    else{

        cout<<"The given number is not prime"<<endl;

        factors(n);

    }

}

**// Q2. Array Operations**

#include <iostream>

using namespace std;

void reverseArray(int arr[], int size) {

    int start = 0, end = size - 1;

    while (start < end) {

        int temp = arr[start];

        arr[start] = arr[end];

        arr[end] = temp;

        start++;

        end--;

    }

}

void findSecondLargestAndSmallest(int arr[], int size) {

    int largest = arr[0], secondLargest = arr[0];

    int smallest = arr[0], secondSmallest = arr[0];

    for (int i = 0; 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];

        }

    }

    cout << "Second Largest Element: " << secondLargest << endl;

    cout << "Second Smallest Element: " << secondSmallest << endl;

}

int main() {

    int size;

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

    cin >> size;

    int arr[size];

    cout << "Enter " << size << " elements for the array: ";

    for (int i = 0; i < size; i++) {

        cin >> arr[i];

    }

    reverseArray(arr, size);

    cout << "Reversed Array: ";

    for (int i = 0; i < size; i++) {

        cout << arr[i] << " ";

    }

    cout << endl;

    findSecondLargestAndSmallest(arr, size);

    return 0;

}

**// Q3. String Manipulation**

#include <iostream>

#include <string>

using namespace std;

bool isPalindrome(string str) {

    int left = 0, right = str.length() - 1;

    while (left < right) {

        char leftChar = str[left];

        char rightChar = str[right];

        if (leftChar >= 'A' && leftChar <= 'Z') {

            leftChar = leftChar + ('a' - 'A');

        }

        if (rightChar >= 'A' && rightChar <= 'Z') {

            rightChar = rightChar + ('a' - 'A');

        }

        if (leftChar != rightChar) {

            return false;

        }

        left++;

        right--;

    }

    return true;

}

void countFrequency(string str) {

    int freq[26] = {0};

    for (int i = 0; i < str.length(); i++) {

        char ch = str[i];

        if (ch >= 'A' && ch <= 'Z') {

            ch = ch + ('a' - 'A');

        }

        if (ch >= 'a' && ch <= 'z') {

            freq[ch - 'a']++;

        }

    }

    for (int i = 0; i < 26; i++) {

        if (freq[i] > 0) {

            cout << char(i + 'a') << ": " << freq[i] << endl;

        }

    }

}

void replaceVowels(string &str, char replacement) {

    for (int i = 0; i < str.length(); i++) {

        char ch = str[i];

        if (ch >= 'A' && ch <= 'Z') {

            ch = ch + ('a' - 'A');

        }

        if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u') {

            str[i] = replacement;

        }

    }

}

int main() {

    string str;

    cout << "Enter a string: ";

    getline(cin, str);

    if (isPalindrome(str)) {

        cout << "The string is a palindrome." << endl;

    } else {

        cout << "The string is not a palindrome." << endl;

    }

    cout << "Character frequency (case insensitive):" << endl;

    countFrequency(str);

    replaceVowels(str, '\*');

    cout << "String after replacing vowels with '\*': " << str << endl;

    return 0;

}

**// Q4. Spiral Number Pattern**

#include <iostream>

using namespace std;

void spiralPrint(int matrix[][100], int n) {

    int num = 1;

    int top = 0, bottom = n - 1, left = 0, right = n - 1;

    int dir = 0;

    while (top <= bottom && left <= right) {

        if (dir == 0) {

            for (int i = left; i <= right; i++) {

                matrix[top][i] = num++;

            }

            top++;

        } else if (dir == 1) {

            for (int i = top; i <= bottom; i++) {

                matrix[i][right] = num++;

            }

            right--;

        } else if (dir == 2) {

            for (int i = right; i >= left; i--) {

                matrix[bottom][i] = num++;

            }

            bottom--;

        } else if (dir == 3) {

            for (int i = bottom; i >= top; i--) {

                matrix[i][left] = num++;

            }

            left++;

        }

        dir = (dir + 1) % 4;

    }

    for (int i = 0; i < n; i++) {

        for (int j = 0; j < n; j++) {

            cout << matrix[i][j] << "\t";

        }

        cout << endl;

    }

}

int main() {

    int n;

    cout << "Enter the size of the matrix (n): ";

    cin >> n;

    int matrix[100][100] = {0};

    spiralPrint(matrix, n);

    return 0;

}

**// Q5.  Rotate Matrix 90 Degrees**

#include <iostream>

using namespace std;

void spiralPrint(int matrix[][100], int n) {

    int num = 1;

    int top = 0, bottom = n - 1, left = 0, right = n - 1;

    int dir = 0;

    while (top <= bottom && left <= right) {

        if (dir == 0) {

            for (int i = left; i <= right; i++) {

                matrix[top][i] = num++;

            }

            top++;

        } else if (dir == 1) {

            for (int i = top; i <= bottom; i++) {

                matrix[i][right] = num++;

            }

            right--;

        } else if (dir == 2) {

            for (int i = right; i >= left; i--) {

                matrix[bottom][i] = num++;

            }

            bottom--;

        } else if (dir == 3) {

            for (int i = bottom; i >= top; i--) {

                matrix[i][left] = num++;

            }

            left++;

        }

        dir = (dir + 1) % 4;

    }

    for (int i = 0; i < n; i++) {

        for (int j = 0; j < n; j++) {

            cout << matrix[i][j] << "\t";

        }

        cout << endl;

    }

}

int main() {

    int n;

    cout << "Enter the size of the matrix (n): ";

    cin >> n;

    int matrix[100][100] = {0};

    spiralPrint(matrix, n);

    return 0;

}