QUESTION-1 (A) #include <iostream>

using namespace std;

int isPrime(int num) {

if (num <= 1) {

return 0; // 1 and negative numbers are not prime

}

for (int i = 2; i \* i <= num; i++) { // Loop from 2 to sqrt(num)

if (num % i == 0) {

return 0; // If divisible by any number, it's not prime

}

}

return 1; // If no divisors are found, it is prime

}

int main() {

int a;

cout << "Enter a positive integer: ";

cin >> a;

if (a <= 0) {

cout << "Please enter a positive integer greater than 0." << endl;

return 0; // Exit if input is invalid

}

if (isPrime(a)) {

cout << a << " is a prime number." << endl;

} else {

cout << a << " is not a prime number." << endl;

}

return 0;

}

(B) #include <iostream>

using namespace std;

int isPrime(int num) {

if (num <= 1) {

return 0;

}

for (int i = 2; i \* i <= num; i++) {

if (num % i == 0) {

return 0;

}

}

return 1;

}

void findFactors(int num) {

cout << "Factors of " << num << " are: ";

for (int i = 1; i <= num; i++) {

if (num % i == 0) {

cout << i << " ";

}

}

cout << endl;

}

int main() {

int a;

cout << "Enter a positive integer: ";

cin >> a;

if (a <= 0) {

cout << "Please enter a positive integer greater than 0." << endl;

return 0;

}

if (isPrime(a)) {

cout << a << " is a prime number." << endl;

} else {

cout << a << " is not a prime number." << endl;

findFactors(a);

}

return 0;

}

(C) #include <iostream>

using namespace std;

int isPrime(int num) {

if (num <= 1) return 0;

for (int i = 2; i \* i <= num; i++) {

if (num % i == 0) return 0;

}

return 1;

}

int nextPrime(int num) {

int next = num + 1;

while (!isPrime(next)) next++;

return next;

}

int main() {

int a;

cout << "Enter a positive integer: ";

cin >> a;

if (a <= 0) {

cout << "Please enter a positive integer greater than 0." << endl;

return 0;

}

if (isPrime(a)) {

cout << "Next prime number greater than " << a << " is " << nextPrime(a) << endl;

} else {

cout << a << " is not a prime number." << endl;

}

return 0;

}

QUESTION-2 (A) #include <iostream>

using namespace std;

int main() {

int n;

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

cin >> n;

int arr[n];

cout << "Enter " << n << " integers:" << endl;

for (int i = 0; i < n; i++) {

cin >> arr[i];

}

cout << "You entered the following array:" << endl;

for (int i = 0; i < n; i++) {

cout << arr[i] << " ";

}

cout << endl;

return 0;

}

(B) #include <iostream>

using namespace std;

int main() {

int n;

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

cin >> n;

int arr[n];

cout << "Enter " << n << " integers:" << endl;

for (int i = 0; i < n; i++) {

cin >> arr[i];

}

for (int i = 0; i < n / 2; i++) {

int temp = arr[i];

arr[i] = arr[n - i - 1];

arr[n - i - 1] = temp;

}

cout << "Reversed array:" << endl;

for (int i = 0; i < n; i++) {

cout << arr[i] << " ";

}

cout << endl;

return 0;

}

(C) #include <iostream>

using namespace std;

int main() {

int n;

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

cin >> n;

if (n < 2) {

cout << "Array must have at least two elements to find second largest and second smallest." << endl;

return 0;

}

int arr[n];

cout << "Enter " << n << " integers:" << endl;

for (int i = 0; i < n; i++) {

cin >> arr[i];

}

int firstLargest = arr[0], secondLargest = -1e9;

int firstSmallest = arr[0], secondSmallest = 1e9;

for (int i = 1; i < n; i++) {

if (arr[i] > firstLargest) {

secondLargest = firstLargest;

firstLargest = arr[i];

} else if (arr[i] > secondLargest && arr[i] != firstLargest) {

secondLargest = arr[i];

}

if (arr[i] < firstSmallest) {

secondSmallest = firstSmallest;

firstSmallest = arr[i];

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

secondSmallest = arr[i];

}

}

if (secondLargest == -1e9) {

cout << "No second largest element." << endl;

} else {

cout << "Second largest element: " << secondLargest << endl;

}

if (secondSmallest == 1e9) {

cout << "No second smallest element." << endl;

} else {

cout << "Second smallest element: " << secondSmallest << endl;

}

return 0;

}

QUESTION-3 (A) #include <iostream>

#include <string>

int main() {

std::string userInput;

std::cout << "Enter a string: ";

std::getline(std::cin, userInput);

std::cout << "You entered: " << userInput << std::endl;

return 0;

}

(B) #include <iostream>

#include <string>

int isPalindrome(const std::string& str) {

int left = 0;

int right = str.length() - 1;

while (left < right) {

if (!isalnum(str[left])) {

left++;

} else if (!isalnum(str[right])) {

right--;

} else {

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 0;

}

left++;

right--;

}

}

return 1;

}

int main() {

std::string input;

std::cout << "Enter a string: ";

std::getline(std::cin, input);

int result = isPalindrome(input);

if (result == 1) {

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

} else {

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

}

return 0;

}

(C) #include <iostream>

#include <string>

int main() {

std::string input;

std::cout << "Enter a string: ";

std::getline(std::cin, input);

int frequency[256] = {0};

for (char ch : input) {

frequency[(unsigned char)ch]++;

}

std::cout << "Character frequencies:" << std::endl;

for (int i = 0; i < 256; i++) {

if (frequency[i] > 0) {

std::cout << "'" << (char)i << "' : " << frequency[i] << std::endl;

}

}

return 0;

}

(D) #include <iostream>

#include <string>

void replaceVowelsWithHash(std::string& str) {

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

if (str[i] == 'a' || str[i] == 'e' || str[i] == 'i' || str[i] == 'o' || str[i] == 'u' ||

str[i] == 'A' || str[i] == 'E' || str[i] == 'I' || str[i] == 'O' || str[i] == 'U') {

str[i] = '#';

}

}

}

int main() {

std::string input;

std::cout << "Enter a string: ";

std::getline(std::cin, input);

replaceVowelsWithHash(input);

std::cout << "Modified string: " << input << std::endl;

return 0;

}

QUESTION-4 #include <iostream>

void printSpiral(int n) {

int matrix[n][n];

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

int num = 1;

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

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

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

}

top++;

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

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

}

right--;

if (top <= bottom) {

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

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

}

bottom--;

}

if (left <= right) {

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

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

}

left++;

}

}

for (int i = 0; i < n; i++) {

for (int j = 0; j < n; j++) {

std::cout << matrix[i][j] << " ";

}

std::cout << std::endl;

}

}

int main() {

int n;

std::cout << "Enter the size of the spiral matrix: ";

std::cin >> n;

printSpiral(n);

return 0;

}

QUESTION-5 #include <iostream>

void rotateMatrix(int matrix[][10], int n) {

for (int i = 0; i < n; i++) {

for (int j = i + 1; j < n; j++) {

std::swap(matrix[i][j], matrix[j][i]);

}

}

for (int i = 0; i < n; i++) {

int left = 0, right = n - 1;

while (left < right) {

std::swap(matrix[i][left], matrix[i][right]);

left++;

right--;

}

}

}

void printMatrix(int matrix[][10], int n) {

for (int i = 0; i < n; i++) {

for (int j = 0; j < n; j++) {

std::cout << matrix[i][j] << " ";

}

std::cout << std::endl;

}

}

int main() {

int n;

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

std::cin >> n;

int matrix[10][10];

std::cout << "Enter the elements of the matrix:" << std::endl;

for (int i = 0; i < n; i++) {

for (int j = 0; j < n; j++) {

std::cin >> matrix[i][j];

}

}

std::cout << "\nOriginal Matrix:" << std::endl;

printMatrix(matrix, n);

rotateMatrix(matrix, n);

std::cout << "\nMatrix after 90-degree rotation:" << std::endl;

printMatrix(matrix, n);

return 0;

}