**PHẦN 1: HÀM**

Bài 1:

int reverseNumber(int num) {

int reversedNumber = 0;

while (num != 0) {

reversedNumber = reversedNumber \* 10 + num % 10;

num /= 10;

}

return reversedNumber;

}

Bài 2:

int lengthOfNumber(int num) {

int count = 0;

while (num != 0) {

count++;

num /= 10;

}

return count;

}

Bài 3:

int fibonacci(int n) {

if (n <= 1) {

return n;

}

return fibonacci(n - 1) + fibonacci(n - 2);

}

Bài 4:

void decimalToBinary(int num) {

int binaryNumber = 0, remainder, count = 1;

while (num != 0) {

remainder = num % 2;

binaryNumber += remainder \* count;

count \*= 10;

num /= 2;

}

printf("So nhi phan cua %d la: %d\n", num, binaryNumber);

}

void decimalToHexadecimal(int num) {

int hexadecimalNumber = 0, remainder, count = 1;

char hexadecimalDigits[] = { '0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'A', 'B', 'C', 'D', 'E', 'F' };

while (num != 0) {

remainder = num % 16;

hexadecimalNumber += remainder \* count;

count \*= 10;

num /= 16;

}

printf("So thap luc phan cua %d la: %X\n", num, hexadecimalNumber);

}

Bài 5:

int nCr(int n, int k) {

if (k > n) {

return 0;

}

if (k == 0 || k == n) {

return 1;

}

return nCr(n - 1, k - 1) + nCr(n - 1, k);

}

Bài 6:

#include <iostream>

using namespace std;

int main() {

int m, n, sum = 0;

cout << "Nhap M va N (M < N): ";

cin >> m >> n;

if (m > n) {

cout << "M phai nho hon N!";

return 0;

}

int a = 0, b = 1, c;

while (m <= n) {

c = a + b;

if (c % 2 == 0) {

sum += c;

}

a = b;

b = c;

m++;

}

cout << "Tong cac so chan trong day Fibonacci tu F(" << m << ") den F(" << n << ") la: " << sum << endl;

return 0;

}

Bài 7:

#include <iostream>

using namespace std;

int gcd(int a, int b) {

while (b) {

a %= b;

swap(a, b);

}

return a;

}

int main() {

int a, b, c, S = 0;

cout << "Nhap a, b, c: ";

cin >> a >> b >> c;

S += gcd(a, b);

S += gcd(b, c);

S += gcd(a, c);

cout << "S = " << S << endl;

return 0;

}

Bài 8:

#include <iostream>

using namespace std;

int gcd(int a, int b) {

while (b) {

a %= b;

swap(a, b);

}

return a;

}

int lcm(int a, int b) {

return (a \* b) / gcd(a, b);

}

int main() {

int a, b, c, S = 0;

cout << "Nhap a, b, c: ";

cin >> a >> b >> c;

S += gcd(a, b);

S += lcm(a, c);

cout << "S = " << S << endl;

return 0;

}

Bài 9:

#include <iostream>

using namespace std;

double calculatePolynomial(double coefficients[], int n, double x) {

double result = 0;

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

result += coefficients[i] \* pow(x, n - i);

}

return result;

}

int main() {

int n;

double x;

cout << "Nhap bac cua da thuc: ";

cin >> n;

double coefficients[n + 1];

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

cout << "Nhap he so a" << i << ": ";

cin >> coefficients[i];

}

cout << "Nhap gia trị x: ";

cin >> x;

double result = calculatePolynomial(coefficients, n, x);

cout << "Gia tri da thuc F(" << x << ") = " << result << endl;

return 0;

}

**PHẦN 2: MẢNG**

**Bài 1:**

a)

bool isAllEven(int a[], int n) {

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

if (a[i] % 2 != 0) {

return false;

}

}

return true;

}

b)

void printPrimeNumbers(int a[], int n) {

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

if (isPrime(a[i])) {

cout << a[i] << " ";

}

}

cout << endl;

}

bool isPrime(int n) {

if (n <= 1) {

return false;

}

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

if (n % i == 0) {

return false;

}

}

return true;

}

c)

bool isSorted(int a[], int n) {

for (int i = 0; i < n - 1; i++) {

if (a[i] > a[i + 1]) {

return false;

}

}

return true;

}

bool isSortedIncreasing(int a[], int n) {

for (int i = 0; i < n - 1; i++) {

if (a[i] > a[i + 1]) {

return false;

}

}

return true;

}

bool isSortedDecreasing(int a[], int n) {

for (int i = 0; i < n - 1; i++) {

if (a[i] < a[i + 1]) {

return false;

}

}

return true;

}

d)

bool hasDuplicate(int a[], int n) {

for (int i = 0; i < n - 1; i++) {

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

if (a[i] == a[j]) {

return true;

}

}

}

return false;

}

void removeDuplicate(int a[], int& n) {

for (int i = 0; i < n - 1; i++) {

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

if (a[i] == a[j]) {

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

a[k] = a[k + 1];

}

n--;

j--;

}

}

}

}

e)

int countEvenNumbers(int a[], int n) {

int countEven = 0;

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

if (a[i] % 2 == 0) {

countEven++;

}

}

return countEven;

}

int countOddNumbers(int a[], int n) {

return n - countEvenNumbers(a, n);

}

void compareEvenOddNumbers(int a[], int n) {

int countEven = countEvenNumbers(a, n);

int countOdd = countOddNumbers(a, n);

if (countEven == countOdd) {

cout << "So luong so chan va so le bang nhau." << endl;

}

else if (countEven > countOdd) {

cout << "So luong so chan nhieu hon so luong so le." << endl;

}

else {

cout << "So luong so le nhieu hon so luong so chan." << endl;

}

}

g)

int findFirstPrimeAfterSquare(int a[], int n) {

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

if (isSquare(a[i - 1]) && isPrime(a[i])) {

return a[i];

}

}

return -1;

}

bool isSquare(int n) {

int sqrtN = sqrt(n);

return sqrtN \* sqrtN == n;

}

h)

void separateNonPrimeNumbers(int a[], int n, int b[]) {

int countNonPrime = 0;

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

if (!isPrime(a[i])) {

b[countNonPrime++] = a[i];

}

}

}

l)

void insertIntoDescendingArray(int a[], int& n, int x) {

int i = n - 1;

while (i >= 0 && x < a[i]) {

a[i + 1] = a[i];

i--;

}

a[i + 1] = x;

n++;

}

m)

bool isSymmetrical(int a[], int n) {

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

if (a[i] != a[n - i - 1]) {

return false;

}

}

return true;

}

n)

bool isSortedIncreasing(int a[], int n) {

for (int i = 0; i < n - 1; i++) {

if (a[i] > a[i + 1]) {

return false;

}

}

return true;

}

**Bài 3:**

a)

#include <iostream>

using namespace std;

const int MAX\_SIZE = 10;

int main() {

int m, n;

int A[MAX\_SIZE][MAX\_SIZE];

cout << "Nhap so hang: ";

cin >> m;

cout << "Nhap so cot: ";

cin >> n;

return 0;

}

void inputArray(int A[][MAX\_SIZE], int m, int n) {

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

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

cout << "Nhap phan tu A[" << i << "][" << j << "]: ";

cin >> A[i][j];

}

}

}

void outputArray(int A[][MAX\_SIZE], int m, int n) {

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

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

cout << A[i][j] << " ";

}

cout << endl;

}

}

b)

int countNegativeNumbers(int A[][MAX\_SIZE], int m, int n) {

int count = 0;

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

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

if (A[i][j] < 0) {

count++;

}

}

}

return count;

}

c)

int findMax(int A[][MAX\_SIZE], int m, int n) {

int max = A[0][0];

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

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

if (A[i][j] > max) {

max = A[i][j];

}

}

}

return max;

}

d)

int findMaxSumRow(int A[][MAX\_SIZE], int m, int n) {

int maxSum = A[0][0];

int row = 0;

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

int sum = 0;

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

sum += A[i][j];

}

if (sum > maxSum) {

maxSum = sum;

row = i;

}

}

return row;

}

**Bài 4:**

a)

#include <iostream>

using namespace std;

const int MAX\_SIZE = 10;

int main() {

int n;

int A[MAX\_SIZE][MAX\_SIZE];

cout << "Nhap so cap cua ma tran: ";

cin >> n;

return 0;

}

void inputMatrix(int A[][MAX\_SIZE], int n) {

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

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

cout << "Nhap phan tu A[" << i << "][" << j << "]: ";

cin >> A[i][j];

}

}

}

void outputMatrix(int A[][MAX\_SIZE], int n) {

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

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

cout << A[i][j] << " ";

}

cout << endl;

}

}

b)

int sumOutsideDiagonal(int A[][MAX\_SIZE], int n) {

int sum = 0;

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

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

if (i != j) {

sum += A[i][j];

}

}

}

return sum;

}

c)

int findMaxOnDiagonal(int A[][MAX\_SIZE], int n) {

int max = A[0][0];

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

if (A[i][i] > max) {

max = A[i][i];

}

}

return max;

}

d)

int countNegativeNumbersOnMinorDiagonal(int A[][MAX\_SIZE], int n) {

int count = 0;

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

if (A[i][n - i - 1] < 0) {

count++;

}

}

return count;

}

e)

int countPrimeNumbers(int A[][MAX\_SIZE], int n) {

int count = 0;

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

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

if (isPrime(A[i][j])) {

count++;

}

}

}

return count;

}

bool isPrime(int n) {

if (n <= 1) {

return false;

}

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

if (n % i == 0) {

return false;

}

}

return true;

}

f)

void findMostFrequentNumbers(int A[][MAX\_SIZE], int n) {

int count[MAX\_SIZE \* MAX\_SIZE] = { 0 };

int maxCount = 0;

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

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

count[A[i][j]]++;

}

}

for (int i = 0; i < MAX\_SIZE \* MAX\_SIZE; i++) {

if (count[i] > maxCount) {

maxCount = count[i];

}

}

for (int i = 0; i < MAX\_SIZE \* MAX\_SIZE; i++) {

if (count[i] == maxCount) {

cout << i << " ";

}

}

cout << endl;

}

**PHẦN 3: CHUỖI**

**Bài 1:**

a)

#include <iostream>

using namespace std;

int main() {

string s;

cout << "Nhap chuoi S: ";

cin >> s;

for (int i = s.length() - 1; i >= 0; i--) {

cout << s[i];

}

cout << endl;

return 0;

}

b)

#include <iostream>

using namespace std;

int main() {

string s;

cout << "Nhap chuoi S: ";

cin >> s;

int count\_letters = 0;

int count\_digits = 0;

for (char c : s) {

if (isalpha(c)) {

count\_letters++;

}

else if (isdigit(c)) {

count\_digits++;

}

}

cout << "So luong chu cai: " << count\_letters << endl;

cout << "So luong chu so: " << count\_digits << endl;

return 0;

}

c)

#include <iostream>

using namespace std;

int main() {

string s;

cout << "Nhap chuoi S: ";

getline(cin, s);

s.erase(remove\_if(s.begin(), s.end(), isspace), s.end());

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

if (i == 0 || s[i - 1] == ' ') {

s[i] = toupper(s[i]);

}

}

cout << "Chuoi S sau khi dinh dang: " << s << endl;

return 0;

}

e)

#include <iostream>

using namespace std;

int main() {

string s;

cout << "Nhap chuoi S: ";

getline(cin, s);

s.erase(remove\_if(s.begin(), s.end(), [](char c) { return !isalpha(c); }), s.end());

cout << "Chuoi S sau khi xoa cac ki tu khong phai la chu cai: " << s << endl;

return 0;

}

f)

#include <iostream>

using namespace std;

int main() {

string s;

cout << "Nhap chuoi S: ";

getline(cin, s);

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

if (isupper(s[i])) {

s[i] = '\*';

}

}

cout << "Chuoi S sau khi thay the cac chu in hoa bang ki tu dau \*: " << s << endl;

return 0;

}

g)

#include <iostream>

using namespace std;

int main() {

string s;

cout << "Nhap chuoi S: ";

getline(cin, s);

s += "Hello";

cout << "Chuoi S sau khi them chuoi \"Hello\": " << s << endl;

return 0;

}

Bài 2:

a)

#include <iostream>

using namespace std;

const int MAX\_SIZE = 50;

struct Student {

string fullName;

string firstName;

string lastName;

};

int main() {

int n;

Student students[MAX\_SIZE];

cout << "Nhap so luong sinh vien: ";

cin >> n;

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

cout << "Nhap ho ten sinh vien thu " << i + 1 << ": ";

cin >> students[i].fullName;

}

return 0;

}

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

int pos = students[i].fullName.find(" ");

if (pos != string::npos) {

students[i].firstName = students[i].fullName.substr(0, pos);

students[i].lastName = students[i].fullName.substr(pos + 1);

}

else {

students[i].firstName = students[i].fullName;

students[i].lastName = "";

}

}

c)

sort(students, students + n, [](const Student& s1, const Student& s2) {

return s1.lastName < s2.lastName;

});

d)

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

if (students[i].firstName == "Nguyen") {

cout << "Sinh vien co ho Nguyen: " << students[i].fullName << endl;

}

}

e)

int maxLen = 0;

int maxIndex = -1;

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

if (students[i].lastName.length() > maxLen) {

maxLen = students[i].lastName.length();

maxIndex = i;

}

}

if (maxIndex != -1) {

cout << "Sinh vien co ten dai nhat: " << students[maxIndex].fullName << endl;

}

else {

cout << "Khong co sinh vien nao co ten." << endl;

}