OOP LAB 02

NAME: Abdul Rahim Dawra RollNO: 24K0633

## TASK # 1

#include <iostream>

using namespace std;

void SwapValues(int \*a, int \*b) {

if (a == b) return;

\*a = \*a ^ \*b;

\*b = \*a ^ \*b;

\*a = \*a ^ \*b;

}

int main() {

int x, y;

cout << "Enter two integers: ";

cin >> x >> y;

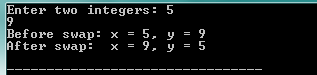
cout << "Before swap: x = " << x << ", y = " << y << endl;

SwapValues(&x, &y);

cout << "After swap: x = " << x << ", y = " << y << endl;

return 0;

}



## TASK # 2

#include <iostream>

#include <string>

using namespace std;

void FirstAndLastIndex(const string &s, char ch, int \*first, int \*last) {

\*first = -1;

\*last = -1;

const char \*p = s.c\_str();

int idx = 0;

while (\*p) {

if (\*p == ch) {

if (\*first == -1) \*first = idx;

\*last = idx;

}

++p;

++idx;

}

}

int main() {

string s;

char ch;

cout << "Enter a string: ";

getline(cin, s);

cout << "Enter a character to find: ";

cin >> ch;

int first, last;

FirstAndLastIndex(s, ch, &first, &last);

if (first == -1) {

cout << "Character '" << ch << "' not found in the string.\n";

} else {

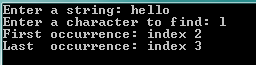
cout << "First occurrence: index " << first << "\n";

cout << "Last occurrence: index " << last << "\n";

}

return 0;

}



## TASK # 3

#include <iostream>

using namespace std;

long long sumArray(int \*arr, int size) {

long long sum = 0;

int \*end = arr + size;

for (int \*p = arr; p < end; ++p) {

sum += \*p;

}

return sum;

}

int main() {

int n;

cout << "Enter number of elements: ";

cin >> n;

int \*arr = new int[n];

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

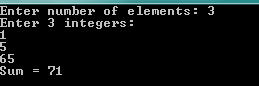
for (int i = 0; i < n; ++i) cin >> arr[i];

cout << "Sum = " << sumArray(arr, n) << endl;

delete[] arr;

return 0;

}



## TASK # 4

#include <iostream>

using namespace std;

int main() {

int N;

cout << "Enter N (matrix size): ";

cin >> N;

if (N <= 0) return 0;

int \*\*mat = new int\*[N];

for (int i = 0; i < N; ++i) mat[i] = new int[N];

cout << "Enter matrix elements row by row (" << N\*N << " values):\n";

for (int i = 0; i < N; ++i)

for (int j = 0; j < N; ++j)

cin >> mat[i][j];

long long mainSum = 0, secSum = 0;

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

mainSum += mat[i][i];

secSum += mat[i][N - i - 1];

}

cout << "\nMatrix:\n";

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

for (int j = 0; j < N; ++j) cout << mat[i][j] << " ";

cout << "\n";

}

cout << "\nMain diagonal sum = " << mainSum << "\n";

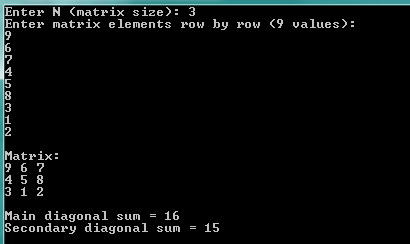
cout << "Secondary diagonal sum = " << secSum << "\n";

for (int i = 0; i < N; ++i) delete[] mat[i];

delete[] mat;

return 0;

}



## TASK # 5

#include <iostream>

#include <cstring>

using namespace std;

int main() {

int len1, len2;

cout << "Enter maximum length for first string: ";

cin >> len1;

cout << "Enter maximum length for second string: ";

cin >> len2;

cin.ignore();

char \*s1 = new char[len1 + 1];

char \*s2 = new char[len2 + 1];

cout << "Enter first string: ";

cin.getline(s1, len1 + 1);

cout << "Enter second string: ";

cin.getline(s2, len2 + 1);

int total = strlen(s1) + strlen(s2);

char \*s3 = new char[total + 1];

strcpy(s3, s1);

strcat(s3, s2);

cout << "\nFirst string: " << s1 << "\n";

cout << "Second string: " << s2 << "\n";

cout << "Concatenated: " << s3 << "\n";

delete[] s1;

delete[] s2;

delete[] s3;

return 0;

}

