**LAB 12 RECURSION TASK 01**

#include <iostream>

using namespace std;

int FUC(int array[], int size){

if (size == 0) {

return 0;

}

else {

return array[0] + FUC(array + 1, size - 1);

}

}

int main() {

int array[] = {1, 2, 3, 4, 5};

int size = sizeof(array) / sizeof(array[0]);

printf("Sum: %d\n", FUC(array, size));

return 0;

}

**LAB 12 RECURSION TASK 02**

#include <iostream>

using namespace std;

#include <string>

bool is\_palindrome(const string& s, int start, int end) {

if (start >= end) {

return true;

}

if (s[start] != s[end]) {

return false;

}

return is\_palindrome(s, start + 1, end - 1);

}

bool is\_palindrome(const string& s) {

string temp;

for (char c : s) {

if (isalnum(c)) {

temp += tolower(c);

}

}

return is\_palindrome(temp, 0, temp.size() - 1);

}

int main() {

cout << boolalpha;

cout << is\_palindrome("Civic") <<endl;

cout << is\_palindrome("Level") <<endl;

cout << is\_palindrome("Hello") <<endl;

return 0;

}