Lr.h

#ifndef lr\_h\_included

#define lr\_h\_included

int lab1();

int lab2();

int lab3();

int lab4();

int lab5();

int lab6();

int lab7();

#endif

Lr.c

#include"lr.h"

#include <stdio.h>

int lab1() {

float V, v1, v2, t;

printf("V =");

scanf("%f", &V);

printf("v1 =");

scanf("%f", &v1);

printf("v2 =");

scanf("%f", &v2);

t = V / (v1 - v2);

printf("искомое время = %f", t);

return 0;

}

int lab2() {

float numerator, denominator, sum;

int k, N;

printf("количество элементов ряда = ");

scanf("%d", &N);

sum = 0;

numerator = 1;

denominator = 1;

k = 1;

while (k <= N)

{

numerator \*= (2 \* k - 1);

denominator \*= (3 \* k - 2);

sum += numerator / denominator;

k++;

}

printf("сумма ряда равна %f", sum);

return 0;

}

int lab3() {

char c;

int count = 0, wordHasA = 0;

printf("Введите текст: ");

while ((c = getchar()) != '\n') {

if (c == ' ' || c == '\t') {

count += wordHasA;

wordHasA = 0;

} else if (c == 'a' || c == 'A') {

wordHasA = 1;

}

}

count += wordHasA;

printf("количество слов, cодержащих букву а: %d\n", count);

return 0;

}

int lab4() {

#include <stdio.h>

int main() {

char word[1000];

char c;

int index = 0;

int is\_first\_char = 1;

char first\_char, last\_char;

printf("Введите текст: ");

while ((c = getchar()) != '\n' && index < 999) {

if (c == ' ' || c == '\t' || c == '\n') {

if (index > 1 && first\_char != last\_char) {

word[index] = '\0';

printf("%s ", word);

}

index = 0;

is\_first\_char = 1;

} else {

if (is\_first\_char) {

first\_char = c;

is\_first\_char = 0;

}

last\_char = c;

word[index++] = c;

}

}

if (index > 1 && first\_char != last\_char) {

word[index] = '\0';

printf("%s", word);

}

return 0;

}

}

int lab5() {

#include <stdio.h>

int main() {

int arr[10];

int minIndex = 0, maxIndex = 0;

// Пользователь вводит числа массива

printf("Введите 10 целых чисел: ");

for (int i = 0; i < 10; i++) {

scanf("%d", &arr[i]);

}

// Находим индексы минимального и максимального элементов

for (int i = 1; i < 10; i++) {

if (arr[i] < arr[minIndex]) minIndex = i;

if (arr[i] > arr[maxIndex]) maxIndex = i;

}

// Обнуляем элементы между минимальным и максимальным

if (minIndex < maxIndex) {

for (int i = minIndex + 1; i < maxIndex; i++) {

arr[i] = 0;

}

} else {

for (int i = maxIndex + 1; i < minIndex; i++) {

arr[i] = 0;

}

}

// Выводим результат

printf("Результат: ");

for (int i = 0; i < 10; i++) {

printf("%d ", arr[i]);

}

return 0;

}

}

int lab6() {

#include <stdio.h>

int main() {

int rows, cols;

printf("Введите количество строк и столбцов: ");

scanf("%d %d", &rows, &cols);

int array[rows][cols];

printf("Введите элементы массива:\n");

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

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

scanf("%d", &array[i][j]);

}

}

//проверка каждой строки на арифметическую прогрессию

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

int isArithmetic = 1; //предположение, что прогрессия

int diff = array[i][1] - array[i][0]; //разность первых двух элементов

for (int j = 2; j < cols && isArithmetic; j++) { //начиная с третьего проверка на совпадение diff

if (array[i][j] - array[i][j-1] != diff) {

isArithmetic = 0;

}

}

if (isArithmetic) {

printf("Строка %d является арифметической прогрессией.\n", i + 1);

} else {

printf("Строка %d не является арифметической прогрессией.\n", i + 1);

}

}

return 0;

}

}

int lab7() {

#include <stdio.h>

int main() {

long long N, K;

scanf("%lld %lld", &N, &K);

char arr[100]; // Запись числа N в строку

sprintf(arr, "%lld", N);

int length = 0; // Длина числа N

while (arr[length] != '\0') {

length++;

}

K %= length; // Сдвиг числа N на K разрядов влево

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

char first = arr[0];

for (int j = 0; j < length - 1; j++) {

arr[j] = arr[j + 1];

}

arr[length - 1] = first;

}

printf("%s\n", arr); // Вывод результата

return 0;

}

main.c

#include"lr.h"

#include <stdio.h>

int main()

{

int choise;

scanf(''%d'', &choise);

switch(choise){

case(1): return lab1(); break;

case(2): return lab2(); break;

case(3): return lab3(); break;

case(4): return lab4(); break;

case(5): return lab5(); break;

case(6): return lab6(); break;

case(7): return lab7(); break;

}

return 0;

}