Министерство образования и науки Кыргызской Республики

Кыргызский государственный технический университет

им.И.Раззакова

Факультет информационных технологий

Кафедра «Программное обеспечение компьютерных систем»

Направление:710400 «Программная инженерия»

ОТЧЕТ

По дисциплине: «Функционально-ориентированное проектирование»

Лабораторная работа №1

Тема: «Функции»

Выполнила: студентка группы

ПИ(б)-5-19 Ажиходжоева Каныкей

Проверил: доцент Искаков Р.Т

Бишкек – 2020

1. Напишите функцию integerPower(base,exponent), которая возвращает значение baseexponent . Например, integerPower(3,4)=3\*3\*3\*3. Предположим, что exponent является положительным ненулевым целым, а base целым. Для управления вычислением функция integerPower(base,exponent) должна применять цикл for. Не используйте никаких функций математической библиотеки.

* Блок-схема



* Код программы

#include<iostream>

using namespace std;

int power(int, int);

int main() {

setlocale(LC\_ALL, "RUS");

int base, res;

int exponent;

cout << "Введите число: ";

cin >> base;

skip:cout << "Введите степень: ";

cin >> exponent;

if (exponent) {

cout << "Введите положительную степень\n";

goto skip;

}

else

res=power(base, exponent);

cout << "Введённое число, в ведённой вами степени равно: " << res<< endl;

system("pause");

return 0;

}

int power(int base, int exponent) {

int result = 1;

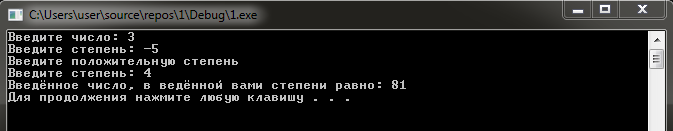
for (int i = 1; i <= exponent; i++) {

result \*= base;

}

return result;}

* Результат вычислений



2. Напишите функцию multiple для двух целых, которая определяет, кратно ли второе число первому. Функция должна получать два целых аргумента и возвращать 1, если второе число кратно первому и 0 в противном случае. Используйте эту функцию в программе, которая вводит серию пар целых чисел.

* Блок-схема



* Код программы

#include<iostream>

#include<ctime>

using namespace std;

bool multiple(int, int);

int main() {

setlocale(LC\_ALL, "RUS");

srand(time(NULL));

int a, b, n;

cout << "Введите колмичество повторений ";

cin >> n;

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

a = rand() % 10 + 1;

b = rand() % 100 + 1;

cout << "Число а равно: " << a << "\t;

cout << "Число b равно: " << b << \t;

if (multiple(a, b))

cout << b << " кратно " << a << endl;

else

cout << b << " не кратно " << a << endl;

}

system("pause");

return 0;

}

bool multiple(int a, int b) {

if (b % a == 0)

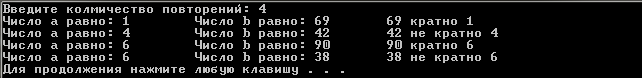
return 1;

else

return 0;

}

* Результат вычислений



3. Напишите функцию, которая выводит у левой границы экрана сплошной квадрат из символов заданного символа (например, \* ), сторона которого определяется целым параметром side. Например, если side равно 4, функция выведет следующее изображение

\* \* \* \*

\* \* \* \*

\* \* \* \*

\* \* \* \*

* Блок-схема



* Код программы

#include<iostream>

using namespace std;

void draw(int s1 = 5, int s2 = 5, char w = '@');

int main() {

setlocale(LC\_ALL, "rus");

int a, b;

char w;

cout << "Введите первую сторону:";

cin >> a;

cout << "Введите вторую сторону:";

cin >> b;

cout << "Введите Символ:";

cin >> w;

draw();

cout << endl;

draw(a, b);

cout << endl;

draw(a, b, w);

system("pause");

return 0;

}

void draw(int s1, int s2, char w) {

for (int i = 0; i < s1; i++) {

for (int j = 0; j < s2; j++) {

cout << w << "\t";

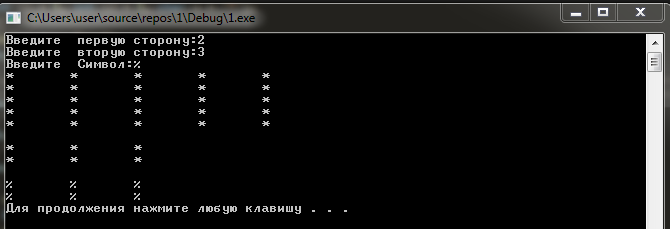
}

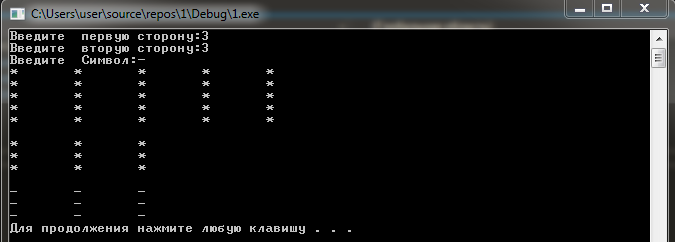
cout << endl;

}

}

* Результат вычислений





4. Даны действительные числа s и t.

Используйте функции и вычислите

 , где 

* Блок-схема



* Код программы

#include<iostream>

using namespace std;

double chislo(double, double, double);

int main() {

setlocale(LC\_ALL, "rus");

double s, t, e, f;

m:cout << "Введите число s: ";

cin >> s;

cout << "Введите число t: ";

cin >> t;

e = chislo(t, -2 \* s, 1.17) + chislo(2.2, t, s - t);

cout << "Результат: " << e <<"\n\n";

goto m;

system("pause");

return 0;

}

double chislo(double a, double b, double c) {

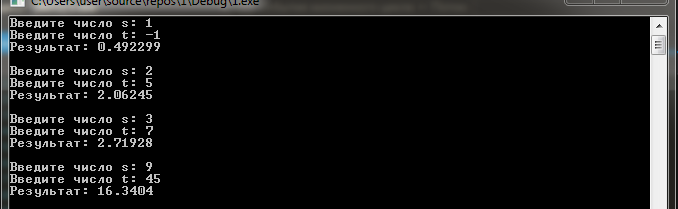
double f;

f = (2 \* a - b - sin(c)) / (5 + abs(c));

return f;

}

* Результат вычислений



5. Даны действительные числа a,b,c. Использовать функции и получить



* Блок-схема



* Код программы

#include<iostream>

using namespace std;

double max(double, double);

int main() {

setlocale(LC\_ALL, "rus");

double a, b, c, f;

cout << "Введите число a: ";

cin >> a;

cout << "Введите число b: ";

cin >> b;

cout << "Введите число c: ";

cin >> c;

f = (max(a, a + b) + max(a, b + c)) / (1 + max(a + b \* c, 1.15));

cout << "Результат: " << f << endl;

system("pause");

return 0;

}

double max(double f, double t) {

if (f > t)

return f;

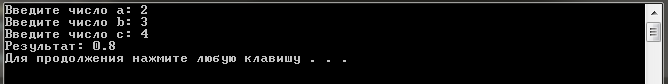
else

return t;

}

* Результат вычислений





6. Напишите программу, которая вводит последовательность целых чисел и передает их по одному функции even, использующей операцию вычисления остатка для определения четности числа. Функция должна принимать целый аргумент и возвращать 1, если аргумент — четное число, и 0 — в противном случае.

* Блок-схема



* Код программы

#include<iostream>

using namespace std;

bool even(int);

int main() {

setlocale(LC\_ALL, "rus");

int a;

do {

cout << "Введите число: ";

cin >> a;

if (a == 0)

break;

if (even(a))

cout << "Число " << a << " - четное" << endl;

else

cout << "Число " << a << " - не четное" << endl;

} while (true);

system("pause");

return 0;

}

bool even(int a) {

if (a % 2 == 0)

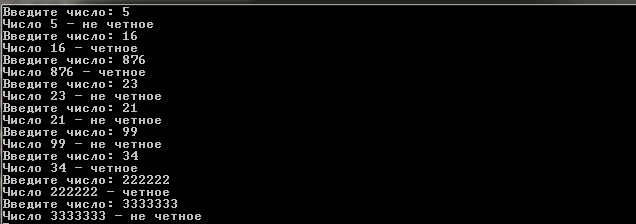
return 1;

else

return 0;

}

* Результат вычислений



7. Говорят, что целое число является совершенным,если его сомножители, включая 1 (но не само число) в сумме дают это число. Например, 6 — это совершенное число, так как 6=1 + 2 + 3. Напишите функцию perfect, которая определяет, является ли параметр number совершенным числом. Используйте эту функцию в программе, которая определяет и печатает все совершенные числа в диапазоне от 1 до 1000. Напечатайте сомножители каждого совер­шенного числа, чтобы убедиться, что число действительно совер­шенное.

* Блок-схема





* Код программы

#include<iostream>

using namespace std;

bool perfect(int);

int main() {

setlocale(LC\_ALL, "rus");

cout << "Cовершенные числа в диапазоне от 1 до 1000:" << endl;

for (int i = 1; i <= 1000; i++) {

if (perfect(i)) {

cout << i << endl;

cout << endl;

cout << "Cомножители: " << endl;

for (int j = 1; j <= i / 2; j++) {

if (i % j == 0) {

cout << j << "+";

}

}

cout << "\n\n";

}

}

system("pause");

return 0;

}

bool perfect(int number) {

int sum = 0;

for (int i = 1; i <= number / 2; i++) {

if (number % i == 0)

sum += i;

}

if (sum == number)

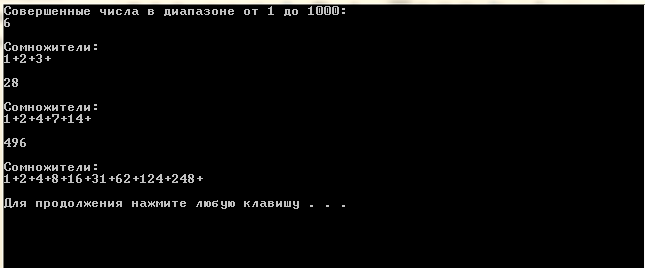
return true;

else

return false;

}

* Результат вычислений



8. Говорят, что целое число является простым числом,если оно, де­лится только на 1 и на само себя. Например, 2, 3, 5 — простые числа, а 4, 6, 8 — нет.

1. Напишите функцию, определяющую, является ли число простым или нет.
2. Используйте эту функцию в программе, которая определяет и печатает все простые числа, лежащие в диапазоне от 1 до 10000.
3. Вначале вы могли бы подумать, что верхней границей, до которой вы должны проводить проверку, чтобы увидеть, является ли число n простым, является n/2, но в действительности вам нужно прове­рить количество чисел, равное корню квадратному из n. Почему? Перепишите программу и запустите ее для обоих способов. Оцените  
   улучшение производительности.

* Блок-схема



* Код программы

#include<iostream>

using namespace std;

bool simple(int);

int main() {

setlocale(LC\_ALL, "rus");

for (int i = 1; i <= 10000; i++)

if (simple(i))

cout << i << "\t";

system("pause");

return 0;

}

bool simple(int num)

{

if (num < 2)

return false;

for (int i = 2; i <= sqrt(num); i++)

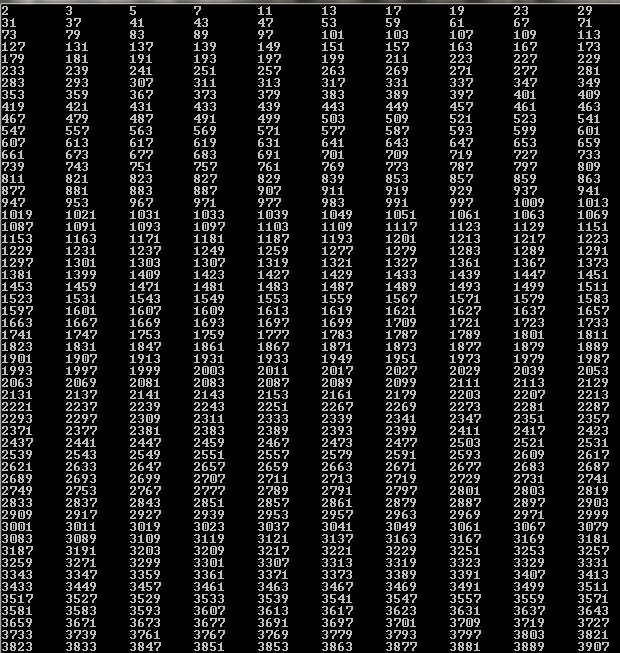
if (num % i == 0)

return false;

return true;

}

* Результат вычислений



9. Напишите программу, моделирующую бросание монеты. Для каж­дого броска монеты программа должна печатать **Орел** или **Решка.** Промоделируйте с помощью этой программы бросание 100 раз и подсчитайте, сколько раз появилась каждая сторона монеты. Напечатайте результаты. Программа должна вызывать отдельную функ­цию **flip,** которая не принимает никаких аргументов и возвращает 0 для **Орла** и 1 для **Решки.**

Замечание: если программа действи­тельно моделирует бросание монеты, каждая сторона монеты должна появляться примерно в половине случаев.

* Блок-схема



* Код программы

#include <iostream>

#include<ctime>

bool flip();

using namespace std;

int main() {

setlocale(LC\_ALL, "rus");

int kolorel = 0, kolreshka = 0;

srand(time(NULL));

for (int i = 1; i <= 100; i++) {

if (flip()) {

cout << i << ".Решка" << "\n";

kolreshka++;

}

else

{

cout << i << ".Орел" << "\n";

kolorel++;

}

}

cout << endl;

cout << "Орел выпал " << kolorel << " раз" << endl;

cout << "Решка выпала " << kolreshka << " раз" << endl;

system("pause");

return 0;

}

bool flip()

{

int a;

a = rand();

if (a % 2 == 0)

return false;

else

return true;

}

* Результат вычислений

