1. Функция

#include <iostream>

//Новые функции и процедуры должны создаваться выше функции main()

//В круглых скобках указываются аргументы, которые будет принимать функция

int sum(int a, int b)

{

//Используем переданные аргументы, чтобы провести вычисления

int result = a + b;

//Оператор возврата говорит программе, что функция завершила работу

return result;

//Всё, что записано после return, не будет выполнено

}

int main()

{

//Вызываем функцию с аргументами

int result = sum(5, 6);

}

1. Процедура

#include <iostream>

void printHello(std::string name)

{

std::cout << "Hello, " << name << "!\n";

}

int main()

{

printHello("Igor");

printHello("Boris");

}

1. Глобальные переменные

#include <iostream>

//Глобальные переменные создаются вне функций

std::string name;

void printHello()

{

std::cout << "Hello, " << name << "!\n";

}

int main()

{

name = "Igor";

printHello();

}

1. Как работают аргументы

#include <iostream>

void sum(int a)

{

a = a + 500;

}

int main()

{

int a = 5;

sum(a);

std::cout << a << "\n";

}

1. Рекурсия

#include <iostream>

int f(int n)

{

if(n < 0)

{

return 0;

}

else if (n == 0)

{

return 1;

}

else

{

return n \* f(n - 1);

}

}

void printFactorial(int n)

{

int a = f(n);

std::cout << n << "! = " << a << "\n";

}

int main()

{

printFactorial(1);

printFactorial(2);

printFactorial(3);

printFactorial(4);

printFactorial(5);

printFactorial(6);

printFactorial(7);

printFactorial(8);

printFactorial(9);

printFactorial(10);

}

Задачи:

1. Программа просит утром у водителя грузового автомобиля ввести следующие данные: средний расход топлива, предполагаемое расстояние которое проедет сегодня автомобиль, стоимость одного литра топлива, остаток топлива в баке. Программа выводит пользователю: предполагаемый остаток топлива в баке, если топлива не хватит на рабочую смену (день), то сумму, которую необходимо взять в кассе для до заправки в кассе, с учетом того, что в баке должно остаться не менее 50 литров в конце смены.

Вечером программа спрашивает, сколько фактически проехал грузовой автомобиль, средний расход топлива по результатам дня, остаток топлива в баке по факту. Расчетный и фактический расход топлива не должен отличаться боле чем на 20%. Если расход топлива отличается более чем на 20%, вывести перерасход.

1. Вывести таблицу умножения от 1х1 до 20х20.
2. Написать рекурсивную процедуру, переводящую числа из одной системы счисления в другую.
3. Нахождение наибольшего общего делителя двух чисел с помощью рекурсивной функции.
4. Реализовать программу в которой присутствует меню выбора действия. Меню должно состоять из вкладок: 1. О программе; 2. Версия программы; 3. Автор; 4 Выход;

Внутри вкладки «О программе», присутствует любая информация или любая программа.

Вкладка «версия программы» выводит вашу версию

Вкладка «Автор» должна содержать фамилию и инициалы, а также электронную почту

Из любой вкладки можно вернуться в главное меню.

Пример считывания клавиш:

#include <conio.h>

while (true)

{

if (\_kbhit())

{

\_getch();

}

}

1. Создать тест на проверку знаний математики, состоящий из пяти вопросов

Задачи выполнить с помощью процедур или функций, можно использовать рекурсию

1. \*Задача вывести матрицу размером NxN, заполненную по спирали от 1 до N2.

Ввод: 5

Вывод:

01 02 03 04 05

16 17 18 19 06

15 24 25 20 07

14 23 22 21 08

13 12 11 10 09

1. \* Пользователь вводит размерность матрицы(таблицы) NxN, программа случайным образом формирует значения в матрице. Программа должна вывести сумму всех элементов матрицы, ниже главной диагонали.

Ввод: 3

Вывод:

1 2 3

4 5 6

7 8 9

Sum: 19

1. \* Каждое натуральное число n,n>1, однозначно раскладывается в произведение простых сомножителей, например, 13=13, 105=3⋅5⋅7, 72=2⋅2⋅2⋅3⋅3. Разложите натуральное число на простые множители (факторизация числа).

\* – задачи повышенной сложности