**Лабораторная №1. Задачи.**

Тема: Программирование. Язык СИ. Математические операции. Переменные и их типы. Операторы. Циклы. Простые условные конструкции. Основы работы со статическими массивами.

Цель: научиться основам программирования на C, освоить условные конструкции, циклы while и for, переменные и их типы, статические массивы.

Оборудование: CLion.

**Комплект 1: начало программирования. Операторы, вычисления, ввод-вывод.**

1.2: написать простую программу. Ввести два числа с клавиатуры, вычислить их сумму и напечатать результат. Использовать функцию printf для приглашений на ввод и для распечатки результата. Использовать функцию scanf для ввода каждого числа отдельно с клавиатуры. Для получения доступа к функциям printf и scanf включить в программу заголовочный файл stdio.h. Использовать корректные спецификаторы форматирования. Здесь и далее для распечатки надписей на экране использовать латинские буквы для избежания проблем с кодировками символов.

#include<stdio.h>  
  
int main() {  
 int a, b;  
 printf("a=");  
 scanf("%d", &a);  
 printf("b=");  
 scanf("%d", &b);  
 printf("a+b=%d", a + b);  
  
 return 0;  
}

Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя переменной | Тип данных | Смысловое обозначение |
| a | int | Переменная для ввода числа с клавиатуры |
| b | int | Переменная для ввода числа с клавиатуры |

Изображение выглядит как снимок экрана, программное обеспечение, Мультимедийное программное обеспечение, дисплей

Контент, сгенерированный ИИ, может содержать ошибки.

1.3: вычислить значение выражения: введя x и y с клавиатуры. Подберите значения аргументов x и y самостоятельно за исключением тривиальных значений. Напечатайте вычисленное значение u (x, y) на экране. Включить в программу заголовочный файл math.h для доступа к математическим функциям.

Изображение выглядит как текст, Шрифт, линия, белый

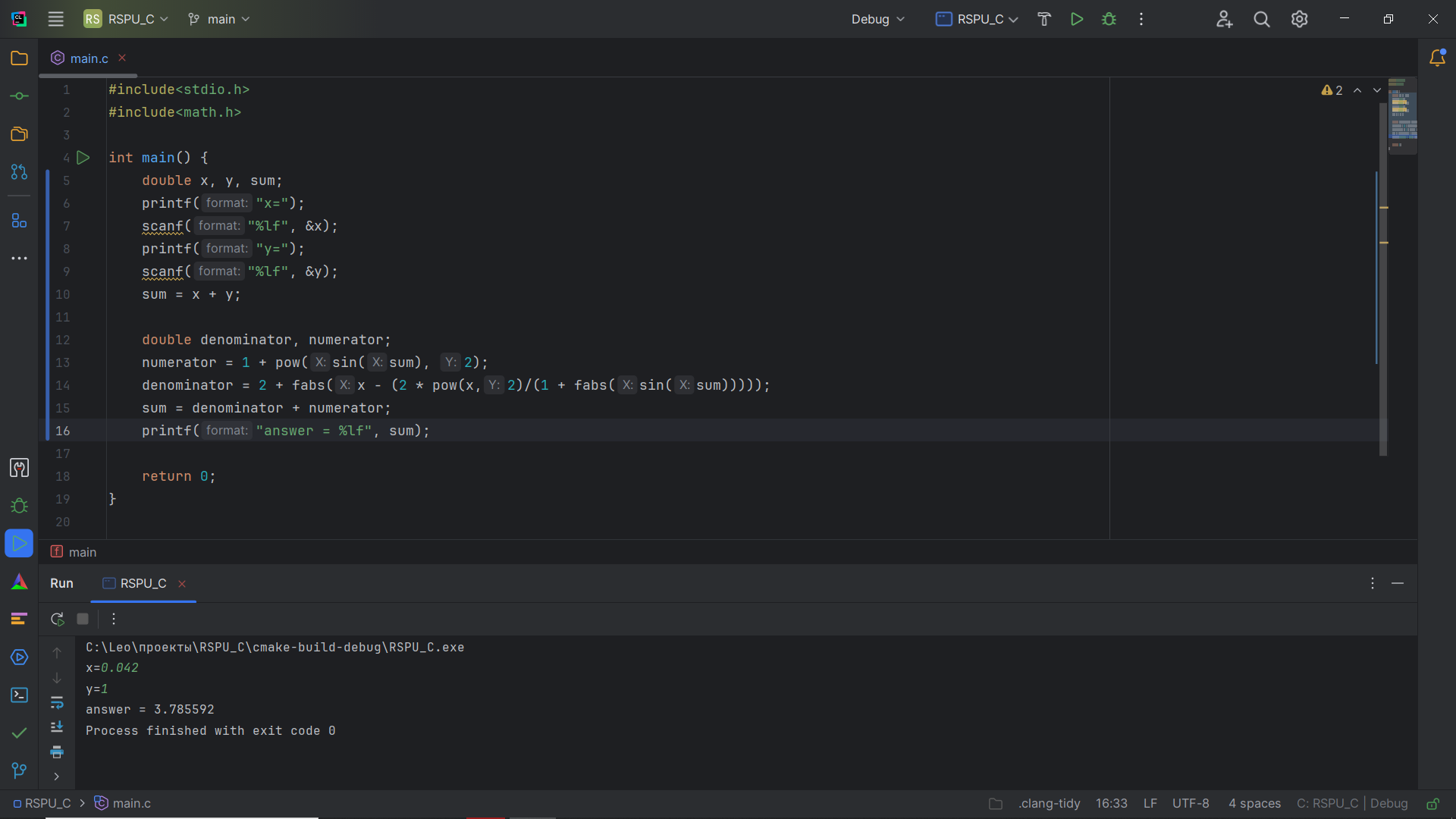
Контент, сгенерированный ИИ, может содержать ошибки.

введя x и y с клавиатуры. Подберите значения аргументов x и y самостоятельно за исключением тривиальных значений. Напечатайте вычисленное значение u (x, y) на экране. Включить в программу заголовочный файл math.h для доступа к математическим функциям.

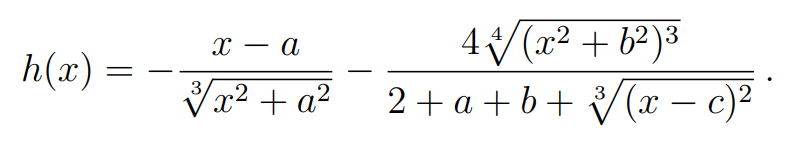
#include<stdio.h>  
#include<math.h>  
  
int main() {  
 double x, y, sum;  
 printf("x=");  
 scanf("%lf", &x);  
 printf("y=");  
 scanf("%lf", &y);  
 sum = x + y;  
  
 double denominator, numerator;  
 numerator = 1 + pow(sin(sum), 2);  
 denominator = 2 + fabs(x - (2 \* pow(x,2)/(1 + fabs(sin(sum)))));  
 sum = denominator + numerator;  
 printf("answer = %lf", sum);  
  
 return 0;  
}

Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя переменной | Тип данных | Смысловое обозначение |
| x | int | Переменная для ввода числа с клавиатуры |
| y | int | Переменная для ввода числа с клавиатуры |
| sum | int | Результирующая переменная |
| denominator | double | Переменная - знаменатель дроби |
| numerator | double | Переменная – числитель дроби |



1.4: вычислить значение выражения:



Выполнить для следующих значений:

1. a = 0.12, b = 3.5, c = 2.4, x = 1.4;
2. a = 0.12, b = 3.5, c = 2.4, x = 1.6;
3. a = 0.27, b = 3.9, c = 2.8, x = 1.8.

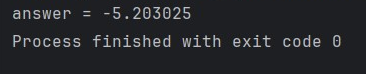
Значения параметров и аргументов можно вводить прямо в коде программы без ввода с клавиатуры.

1. answer = -5.203025
2. answer = -5.235003
3. answer = -5.619954

#include<stdio.h>  
#include<math.h>  
  
int main() {  
 double a, b, c, x;  
 a = 0.13;  
 b = 3.5;  
 c = 2.4;  
 x = 1.4;  
  
 double fraction1, denominator1, numerator1;  
 denominator1 = pow(pow(x,2)+pow(a,2), 1/3);  
 numerator1 = x - a;  
 fraction1 = denominator1 / numerator1;  
  
 double fraction2, denomination2, numerator2;  
 numerator2 = 4 \* pow(((x\*x)+(b\*b)),0.75);  
 denomination2 = 2 + a + b + pow((x-c),(2/3));  
 fraction2 = numerator2 / denomination2;  
 printf("answer = %lf", (-fraction1 - fraction2));  
  
 return 0;  
}

Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя переменной | Тип данных | Смысловое обозначение |
| a | double | Переменная для ввода числа с клавиатуры |
| b | double | Переменная для ввода числа с клавиатуры |
| c | double | Переменная для ввода числа с клавиатуры |
| x | double | Переменная для ввода числа с клавиатуры |
| fraction1 | double | Результирующая переменная первой дроби |
| denominator1 | double | Результирующая переменная знаменателя первой дроби |
| numerator1 | double | Результирующая переменная числителя первой дроби |
| fraction2 | double | Результирующая переменная второй дроби |
| denominator2 | double | Результирующая переменная знаменателя второй дроби |
| numerator2 | double | Результирующая переменная числителя второй дроби |

****

**Комплект 2: Организация циклов. Условные конструкции.**

2.1: вычислить, используя цикл for. Распечатать на экране координаты для каждой итерации по t. Координаты планеты Марс для каждой итерации задаются заданы формулами:

Изображение выглядит как текст, Шрифт, снимок экрана, линия

Контент, сгенерированный ИИ, может содержать ошибки.

где r1 – радиус орбиты Марса, r2 – радиус орбиты Земли, T1 и T2 — периоды обращения указанных планет соответственно, t – каждый заданный момент времени внутри цикла по времени. Подберите подходящие единицы измерения для времени и расстояния.

#include<stdio.h>  
#include<math.h>  
  
int main() {  
 double x, y, r1, r2, T1, T2, W1, W2;  
 //еденицы имзерения в СИ  
 r1 = 3396000; //радиус орбиты Марса (м)  
 r2 = 6371000; //радиус орбиты Земли (м)  
 T1 = 59355072; //период обращения Марса вокруг Солнца (с)  
 T2 = 31536000; //период обращения Земли вокруг Солнца (с)  
 W1 = 2\*M\_PI/T1;  
 W2 = 2\*M\_PI/T2;  
   
 for (int t = 0; t <= 1; t++){  
 x = ((r1 \* cos(W1\*t)) - (r2 \* cos(W2\*t)));  
 y = ((r1 \* sin(W1\*t)) - (r2 \* sin(W2\*t)));  
 printf("t=%d x=%lf y=%lf\n", t, x, y);  
 }  
  
 return 0;  
}

Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя переменной | Тип данных | Смысловое обозначение |
| x | double | Координаты Марса относительно Земли по X |
| y | double | Координаты Марса относительно Земли по Y |
| r1 | double | радиус орбиты Марса в метрах |
| r2 | double | радиус орбиты Земли в метрах |
| Т1 | double | период обращения Марса вокруг Солнца в секундах |
| Т2 | double | период обращения Земли вокруг Солнца в секундах |
| W1 | double | Частота вращения Марса |
| W2 | double | Частота вращения Земли |
| t | int | Переменная-счётчик в цикле for / время в секундах |

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

**2.2: вычислить определённый интеграл от заданной функции методом трапеций:**

**Изображение выглядит как Шрифт, рукописный текст, текст, линия

Контент, сгенерированный ИИ, может содержать ошибки.**

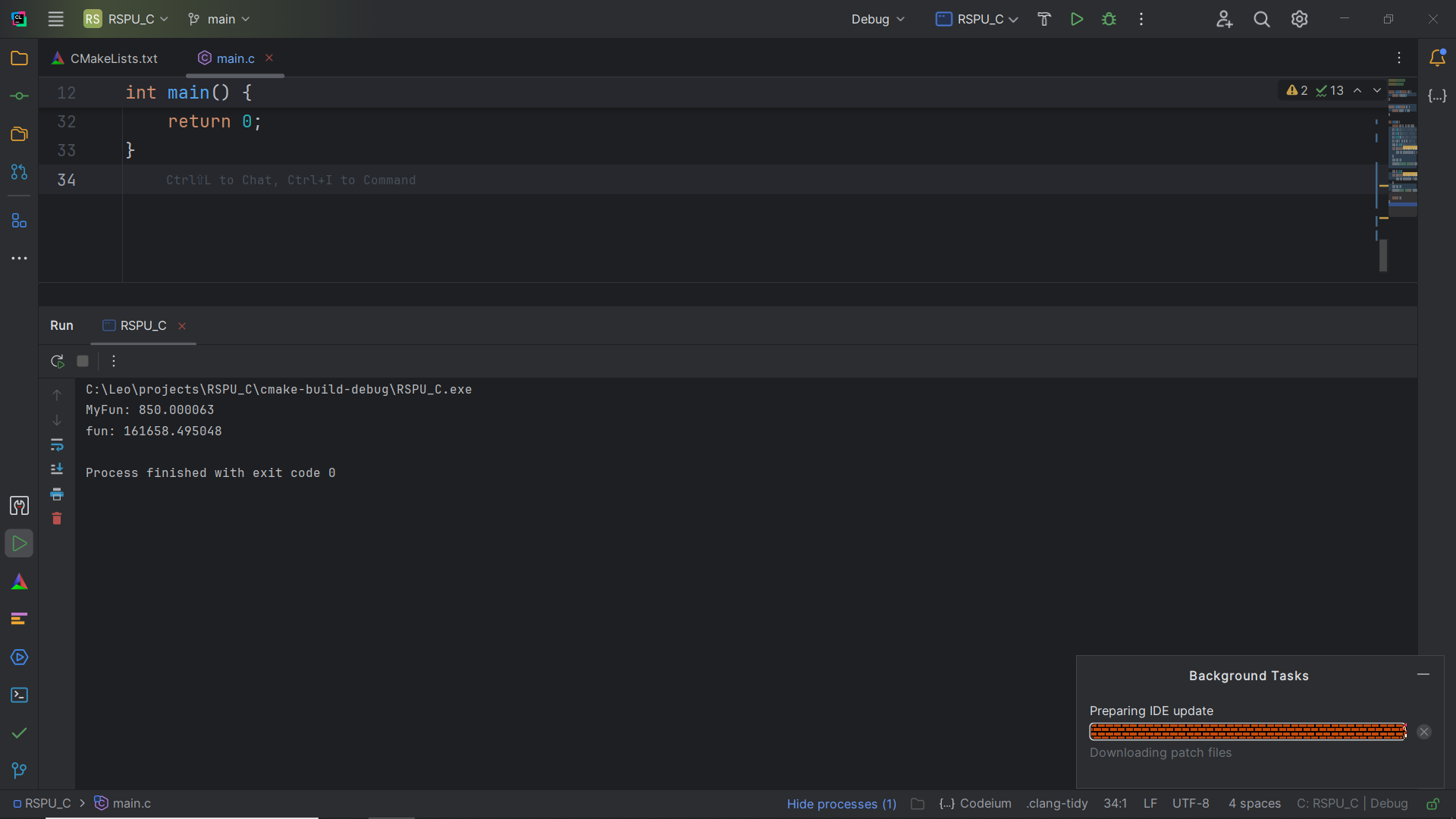
Функция f(x) может быть выбрана и самостоятельно. Результат интегрирования сравнить с вычисленным вручную и убедиться в корректности результата.

*Выбранная функция*: f(x) = 3x2-5

#include<stdio.h>  
#include<math.h>  
  
double MyFun(double x) {  
 return 3\*x\*x-5;  
}  
  
double fun(double x) {  
 return exp(x + 2);  
}  
  
int main() {  
 double a, b, n, h, sum;  
 a = 5; //левая граница интервала  
 b = 10; //правая граница интервала  
 n = 1000; //кол-во интервалов между a и b  
 h = (b - a) / n; //длина каждого интервала  
 sum = 0.0; //переменная-ответ  
 for (double i = a; i <= b; i += h) {  
 sum += (MyFun(i+h) + MyFun(i))\*h;  
 }  
 sum /= 2;  
 printf("MyFun: %lf\n", sum);  
  
 sum = 0.0;  
 for (double i = a; i <= b; i += h) {  
 sum += (fun(i+h) + fun(i))\*h;  
 }  
 sum /= 2;  
 printf("fun: %lf\n", sum);  
  
 return 0;  
}

Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя переменной | Тип данных | Смысловое обозначение |
| x | double | Переменная аргумент в функциях: MyFun и fun |
| a | double | Левая граница интервала |
| b | double | Правая граница интервала |
| n | double | Кол-во интервалов между a, b |
| h | double | Длина каждого интервала |
| sum | double | Результирующая переменная |
| i | double | Переменная-счётчик в цикле |



2.3: организовать и распечатать последовательность чисел Падована, не превосходящих число m, введенное с клавиатуры. Числа Падована представлены следующим рядом: 1, 1, 1, 2, 2, 3, 4, 5, 7, 9, 12, 16, 21, 28, 37, 49, 65, 86, 114, 151, 200, 265, ... Использовать конструкцию for и простые варианты условной конструкции if else. Для этих чисел заданы формулы:

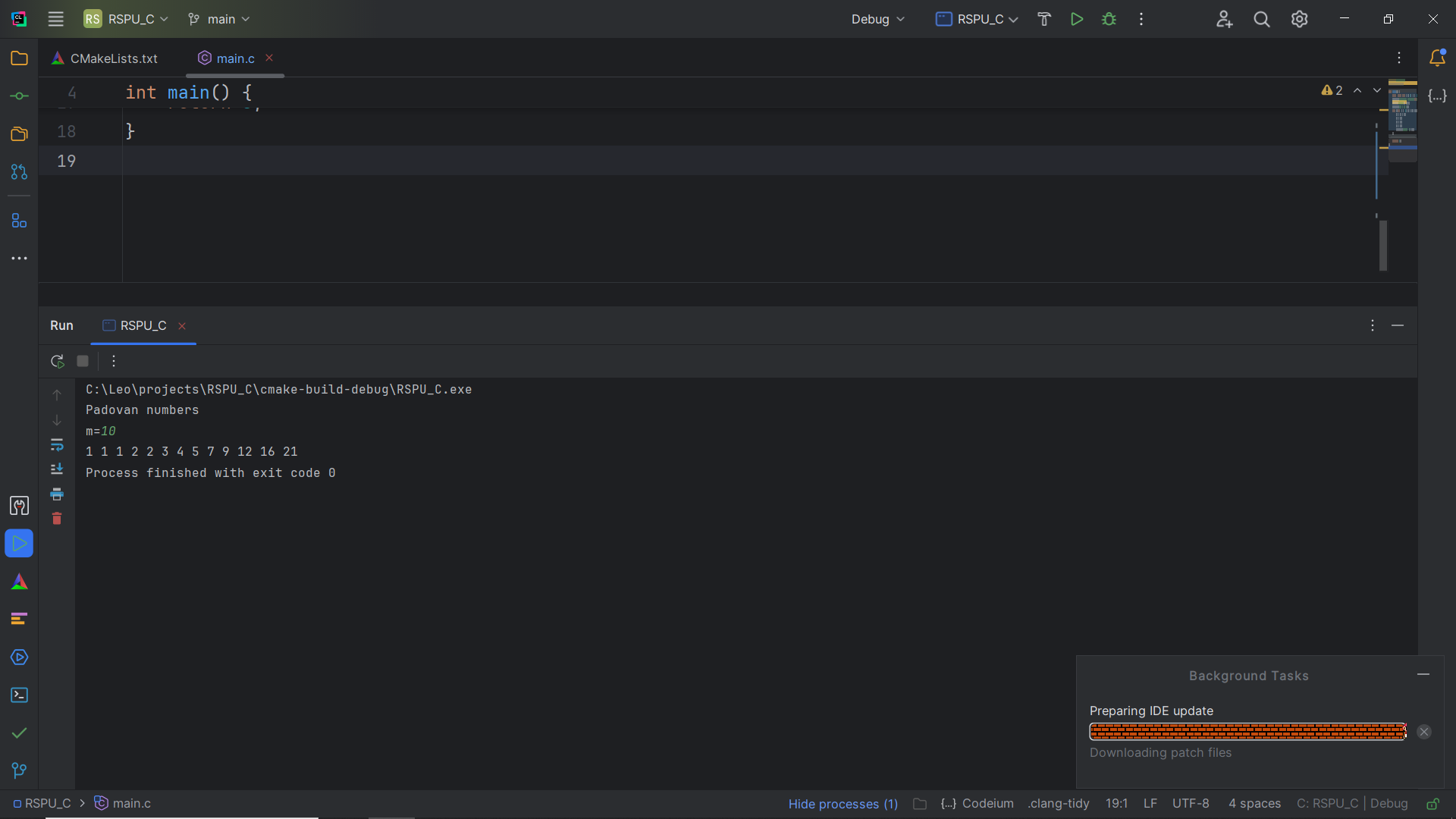
**Изображение выглядит как Шрифт, текст, белый, типография

Контент, сгенерированный ИИ, может содержать ошибки.**

#include<stdio.h>  
#include<math.h>  
  
int main() {  
 long long int a = 1, b = 1, c = 1, d, m;  
 printf("Padovan numbers\nm=");  
 scanf("%lld", &m);  
 printf("1 1 1 ");  
 for (int i = 0; i < m; i++) {  
 d = a + b;  
 a = b;  
 b = c;  
 c = d;  
 printf("%lld ", d);  
 }  
   
 return 0;  
}

Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя переменной | Тип данных | Смысловое обозначение |
| a | long long int | Член последовательности чисел Падована |
| b | long long int | Член последовательности чисел Падована |
| c | long long int | Член последовательности чисел Падована |
| d | long long int | Член последовательности чисел Падована |
| m | long long int | Кол-во членов последовательности чисел Падована |
| i | int | Переменная-счётчик в цикле |



2.4: С клавиатуры вводится трёхзначное число, считается сумма его цифр. Если сумма цифр числа больше 10, то вводится следующее трёхзначное число, если сумма меньше либо равна 10 — программа завершается.

#include<stdio.h>  
#include<math.h>  
  
int main() {  
 int x, sum = 11;  
 while (sum > 10){  
 printf("x=");  
 scanf("%d", &x);  
 sum = 0;  
 for (int i = 0; i < 3; i++){  
 sum += x % 10;  
 x /= 10;  
 }  
 printf("sum=%d\n", sum);  
 }  
   
 return 0;  
}

Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя переменной | Тип данных | Смысловое обозначение |
| x | int | 3-х значное число |
| sum | int | Сумма трёх цифр числа |

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

**Комплект 3: Основы работы со статическими массивами.**

3.1: для некоторого числового вектора X, введённого с клавиатуры, вычислить значения вектора

Y = X · X (yi = xi · xi — поэлементно).

#include<stdio.h>  
#include<math.h>  
  
int main() {  
 const int SIZE = 5;  
 int x[SIZE], y[SIZE];  
 printf("Enter five numbers:");  
 for (int i = 0; i < SIZE; i++){  
 scanf("%d", &x[i]);  
 y[i] = x[i]\*x[i];  
 }  
 printf("new vector:");  
 for (int i = 0; i < SIZE; i++){  
 printf("%d ", y[i]);  
 }  
 printf("\n");  
  
 return 0;  
}

Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя переменной | Тип данных | Смысловое обозначение |
| SIZE | int | Размер массива |
| x | Int[] | Массив размера SIZE |
| y | Int[] | Массив размера SIZE |
| i | Int | Переменная-счётчик в цикле |

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

3.2: для некоторого числового массива X, введённого с клавиатуры поэлементно, изменить порядок элементов на обратный и распечатать результат на экране.

#include<stdio.h>  
#include<math.h>  
  
int main() {  
 const int SIZE = 5;  
 int x[SIZE], c;  
 printf("Enter vector (%d):", SIZE);  
 for (int i = 0; i < SIZE; i++){  
 scanf("%d", &x[i]);  
 }  
  
 for (int i = 0; i < SIZE / 2; i++){  
 c = x[i];  
 x[i] = x[SIZE-i-1];  
 x[SIZE-i-1]=c;  
 }  
  
 printf("new vector:");  
 for (int i = 0; i < SIZE; i++){  
 printf("%d ", x[i]);  
 }  
 printf("\n");  
  
 return 0;  
}

Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя переменной | Тип данных | Смысловое обозначение |
| SIZE | int | Размер массива |
| x | Int[] | Массив размера SIZE |
| с | Int | Переменная стакан |
| i | Int | Переменная-счётчик в цикле |

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

3.3: транспонировать матрицу:

Изображение выглядит как Шрифт, число, линия, типография

Контент, сгенерированный ИИ, может содержать ошибки.

#include<stdio.h>  
#include<math.h>  
  
int main() {  
 int matrix[3][3] = {{1,2,3},  
 {4,5,6},  
 {7,8,9}};  
 int c;  
  
 //i - строка, g - столбец  
 for (int i = 0; i < 3; i++){  
 for (int g = 0+i; g < 3; g++){  
 c = matrix[i][g];  
 matrix[i][g] = matrix[g][i];  
 matrix[g][i] = c;  
 }  
 }  
  
 for (int i = 0; i < 3; i++){  
 for (int g = 0; g < 3; g++){  
 printf("%d ", matrix[i][g]);  
 }  
 printf("\n");  
 }  
   
 return 0;  
}

Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя переменной | Тип данных | Смысловое обозначение |
| matrix | Int[] | Массив размера SIZE |
| i | Int | Строка матрицы |
| g | Int | Столбец матрицы |
| с | Int | Переменная стакан |

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

3.4: преобразовать исходную матрицу так, чтобы первый элемент каждой строки был заменён средним арифметическим элементов этой строки.

#include<stdio.h>  
#include<math.h>  
  
int main() {  
 double matrix[3][3] = {{1,2,3},  
 {4,5,6},  
 {7,8,9}};  
 double c;  
  
 for (int i = 0; i < 3; i++){  
 c = 0.0;  
 for (int g = 0; g < 3; g++){  
 c += matrix[i][g];  
 }  
 matrix[i][0] = c/3; //среднее арифметическое  
 }  
  
  
 for (int i = 0; i < 3; i++){  
 for (int g = 0; g < 3; g++){  
 printf("%.1lf ", matrix[i][g]);  
 }  
 printf("\n");  
 }  
  
 return 0;  
}

Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя переменной | Тип данных | Смысловое обозначение |
| matrix | Int[] | Массив размера SIZE |
| i | Int | Строка матрицы |
| g | Int | Столбец матрицы |
| с | Int | Переменная стакан |

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

3.5: реализовать самостоятельно алгоритм сортировки вставками (без создания своих функций, внутри функции main).

#include<stdio.h>  
#include<math.h>  
  
int main() {  
 const int SIZE = 10;  
 int a[] = {5, 2, 8, 1, 3, 6, 4, 7, 9, 0};  
 for (int i = 0; i < SIZE; i++)  
 {  
 printf("%d ", a[i]);  
 }  
 printf("\n");  
  
 int temp;  
 for (int i = 0; i < SIZE; i++)  
 {  
 for (int j = 0; j < SIZE - i - 1; j++)  
 {  
 if (a[j] > a[j + 1])  
 {  
 temp = a[j];  
 a[j] = a[j + 1];  
 a[j + 1] = temp;  
 }  
 }  
 }  
  
 for (int i = 0; i < SIZE; i++)  
 {  
 printf("%d ", a[i]);  
 }  
 printf("\n");  
  
 return 0;  
}

Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя переменной | Тип данных | Смысловое обозначение |
| SIZE | int | Размер массива |
| a | Int[] | Массив размера SIZE |
| temp | Int | Переменная стакан |
| i | Int | Переменная-счётчик в цикле |
| j | Int | Переменная-счётчик в цикле |

