Лабораторная работа №9

Задача 1

Постановка задачи: Написать программу для вычисления факториала числа, введенного с клавиатуры.

Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| **Идентификатор** | **Семантика** | **Тип** |
| n | число | int |
| fact | функция вычисления факториала | int |

Код программы:

#include <stdio.h>

#include <stdlib.h>

int fact(int number) {

if (number == 0)

return 1;

else

return fact(number - 1) \* number;

}

int main(int argc, char const \*argv[]) {

int n;

printf("Факториал какого числа нужно вычислить?\n");

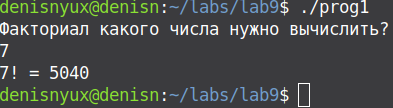
scanf("%d", &n);

printf("%d! = %d\n", n, fact(n));

return 0;

}

Результат выполненной работы:



Задание 2

Постановка задачи: Написать программу для вычисления чисел Фибоначчи

Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| **Идентификатор** | **Семантика** | **Тип** |
| n | число, количество вычисленных чисел фибоначчи | int |
| fib | функция вычисления чисел фибоначчи | int |

Код программы:

## #include <stdio.h>

## #include <stdlib.h>

## 

## int fib(int number) {

## if ((number == 1) || (number == 2)) {

## return 1;

## }

## return fib(number - 1) + fib(number - 2);

## }

## 

## int main(int argc, char const \*argv[]) {

## int n;

## printf("Сколько чисел Фибоначчи нужно вычислить?\n");

## scanf("%d", &n);

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

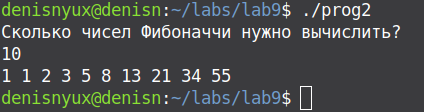
## printf("%d ", fib(i));

## printf("\n");

## return 0;

## }

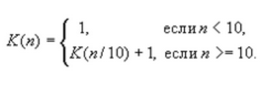
Результат выполненной работы:



Задание 3

Постановка задачи: Определите функцию K(n), которая возвращает количество цифр в заданном натуральном числе n

Математическая модель:



Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| **Идентификатор** | **Семантика** | **Тип** |
| n | число | int |
| amount | функция, считающая количество цифр | int |

Код программы:

## #include <stdio.h>

## #include <stdlib.h>

## 

## int amount(int number) {

## if (number < 10)

## return 1;

## else

## return amount(number / 10) + 1;

## }

## 

## int main(int argc, char const \*argv[]) {

## int n;

## printf("Введите число:\n");

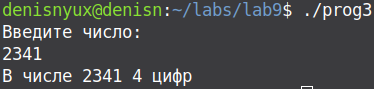
## scanf("%d", &n);

## printf("В числе %d %d цифр\n", n, amount(n));

## return 0;

## }

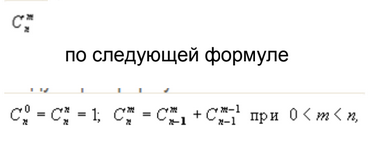
Результат выполненной работы:



Задание 4

Постановка задачи: Функция C(m, n), где 0 <= m <= n, для вычисления биномиального коэффициента.

Математическая модель:



Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| **Идентификатор** | **Семантика** | **Тип** |
| n, m | числа, чей биномиальный коэффициент мы будем считать | int |
| С | функция, считающая коэффициент | int |

Код программы:

## #include <stdio.h>

## #include <stdlib.h>

## 

## int C(int m, int n) {

## if (m == 0 || m == n)

## return 1;

## else

## return C(m, n - 1) + C(m - 1, n - 1);

## }

## 

## int main(int argc, char const \*argv[]) {

## int n, m;

## printf("Введите числа m и n, 0 <= m <= n:\n");

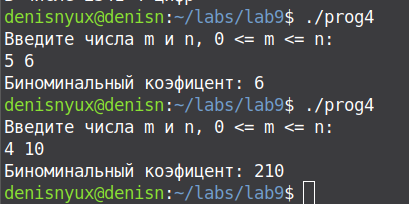
## scanf("%d %d", &m, &n);

## printf("Биноминальный коэфицент: %d\n", C(m, n));

## return 0;

## }

Результат выполненной работы:



Задание 5

Постановка задачи: Вычислить сумму элементов одномерного массива.

Математическая модель: При решении задачи используйте следующее соображение: сумма равна нулю, если количество элементов равно нулю, и сумме всех предыдущих элементов плюс последний, если количество элементов не равно нулю.

Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| **Идентификатор** | **Семантика** | **Тип** |
| n | количество элементов в массиве | int |
| max\_range | верхняя граница диапазона случайных чисел | int |
| void | функция заполнения массива случайными числами | - |
| sum | функция нахождения суммы элементов массива | int |
| a[n] | массив целых чисел | int |

Код программы:

## #include <stdio.h>

## #include <stdlib.h>

## #include <time.h>

## /\*Размер массива\*/

## #define n 8

## /\*Верхняя граница диапазона случайных чисел\*/

## #define max\_range 20

## 

## /\*Функция заполения массива случайными числами\*/

## void filling(int \*array, int amount\_of\_elements) {

## for (int i = 0; i < amount\_of\_elements; i++) {

## array[i] = rand() % max\_range;

## }

## }

## /\*Функция нахождения суммы элементов массива\*/

## int sum(int \*array, int element) {

## if (element == 0) // Кол-во элементов равно 0

## return 0;

## else

## return sum(array, element - 1) + array[element - 1];

## }

## 

## int main(int argc, char const \*argv[]) {

## int a[n];

## filling(a, n);

## printf("Массив А:\n");

## for (int i = 0; i < n; i++) {

## printf("%d ", a[i]);

## }

## printf("\n%d - сумма элементов\n", sum(a, n));

## 

## return 0;

## }

Результат выполненной работы:

