



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»  
КАФЕДРА МЕДИЧНОЇ КІБЕРНЕТИКИ ТА ТЕЛЕМЕДИЦИНИ

## **Лабораторна робота № 9**

з дисципліни «Алгоритмізація та програмування»

на тему: “ Рекурсія ”

Варіант 16

### **Виконав:**

Студент групи БС-32

Хоміщевич Микита

### **Перевірила:**

викл.

Матвійчук А.О.

## Задание

16

Рекурсивно обчислити добуток  $n \geq 2$  співмножників ( $n$  парне):

$$y = \frac{2}{1} \cdot \frac{2}{3} \cdot \frac{4}{3} \cdot \frac{4}{5} \cdot \frac{6}{5} \cdot \frac{6}{7} \cdot \dots$$

☐ Лабораторна робота виконана без зауважень

☐ Лабораторна робота має зауваження:

☐ присутні зауваження до блок-схем:

☐ виконані не за стандартом

☐ не відповідають коду

☐ присутній код

☐ інші зауваження:

☐ присутні зауваження до коду:

☐ задача завдання вирішена хибно

☐ код програми не компілюється

☐ використано глобальні змінні

☐ недостатня декомпозиція на функціонуванні користувача

☐ статичні змінні при роботі з масивами

☐ оформлення коду

☐ присутні зайві символи «{» та «}»

☐ інші зауваження:

☐ невірні відповіді на запитання:

☐ №1 ☐ №2 ☐ №3

☐ мають інші зауваження:

## Программный код

```
#include <iostream>
#include <cstdio>
#include <cmath>
#include <conio.h>
#include <locale.h>

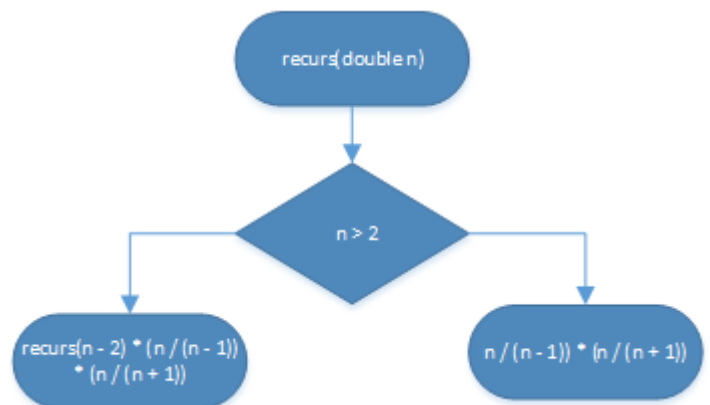
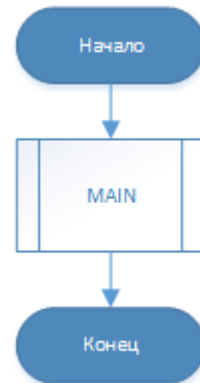
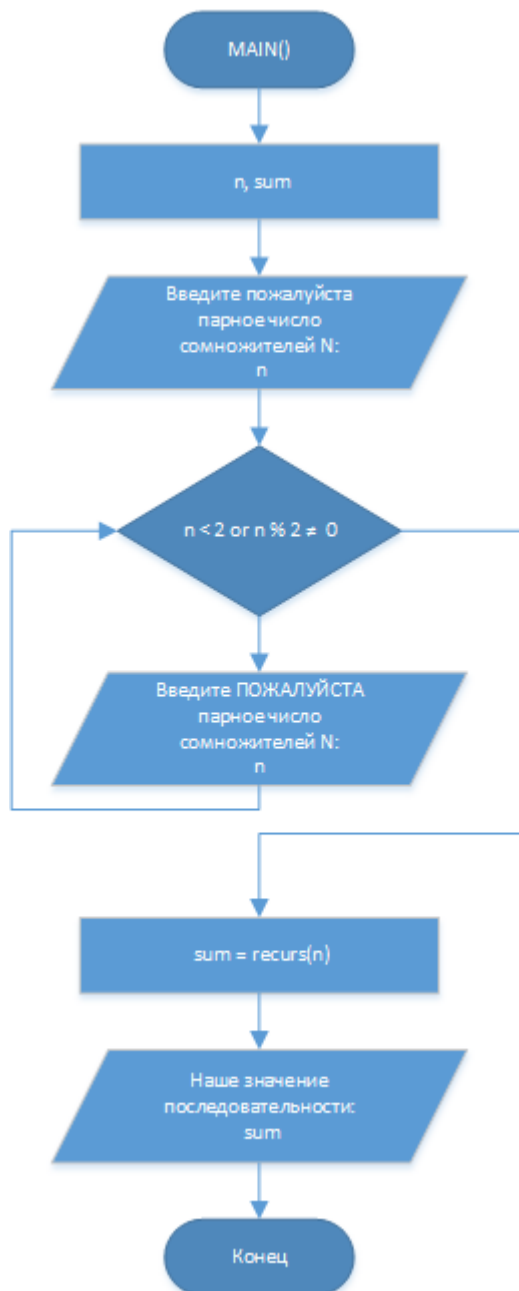
void MAIN();
double recurs(double n);

int main() {
    MAIN();
    _getch();
}

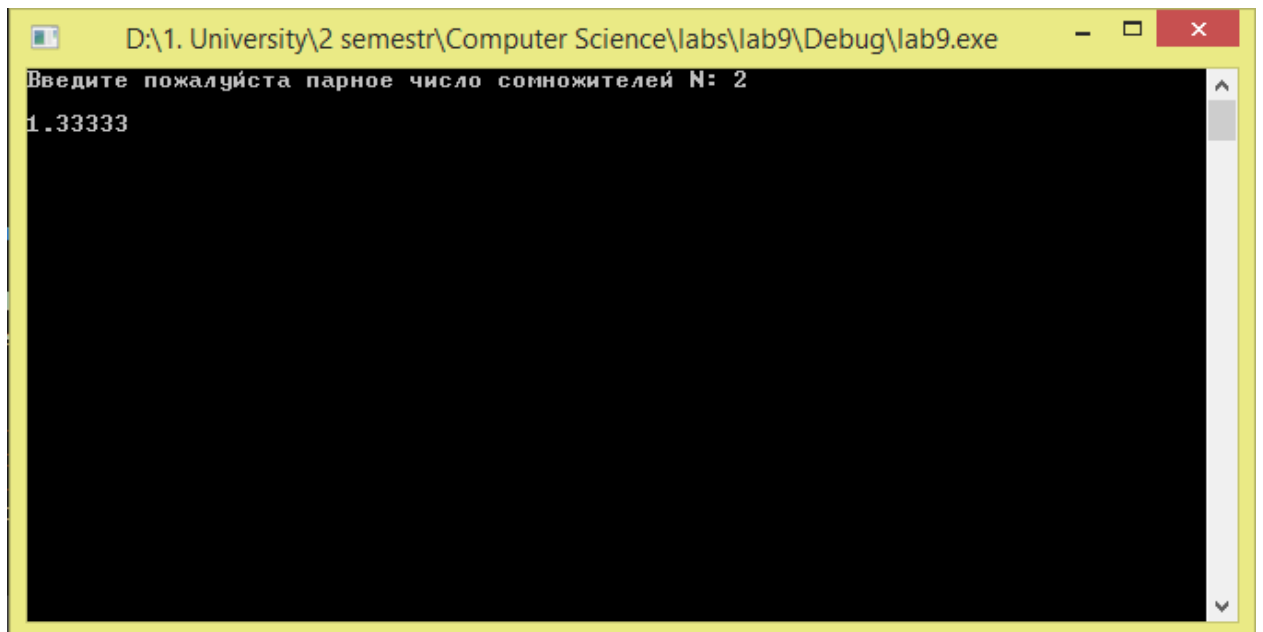
void MAIN() {
    setlocale(LC_ALL, "RUS");
    double n, sum;
    std::cout << "Введите пожалуйста парное число сомножителей N: ";
    std::cin >> n;
    while ((int)n < 2 || ((int)n % 2 != 0)) {
        std::cout << endl << "Введите ПОЖАЛУЙСТА парное число
сомножителей N: ";
        std::cin >> n;
    }
    sum = recurs(n);
    std::cout << "Наше значение последовательности: " << sum << std::endl;
}
```

```
double recurs(double n) {  
    if (n > 2) return (recurs(n - 2) * (n / (n - 1)) * (n / (n + 1)));  
    else return (n / (n - 1)) * (n / (n + 1));  
}
```

## Блок-схема



## Скриншот работы программы



## Ответы на контрольные вопросы

1. Дайте определение рекурсии и приведите примеры ее использования в разных отраслях знаний?

Рекурсия - вызов функции (процедуры) из неё же самой, непосредственно (*простая рекурсия*) или через другие функции (*сложная* или *косвенная рекурсия*), например, функция *A* вызывает функцию *B*, а функция *B* — функцию *A*.

К примеру, в физике рекурсией может обозначиться взаимное отображение двух стекл, что стоят друг на против друга и каждое отражает другое само в себе.

2. В каких задачах разумно использовать рекурсивные функции?

Это зависит от специфики самой задачи. Как правило, любой код можно оформить в виде прохода циклом (итеративно), что более приемлемо для понимания и меньше затратно по памяти. Но если необходимо где-то специально сократить код или рекурсивный вариант легче, чем итеративный, то лучше использовать рекурсию.

3. Что общего и в чем разница между циклическими и рекурсивными способами организации расчетов?

Схожесть:

- 1) Количество шагов (вложенность) для одной задачи одинакова
- 2) При наличии хвостовой рекурсии (константного значения выделяемой памяти компилятором) кол-во итераций в цикле и кол-во вызовов рекурсии будет одинаково

Различия:

- 1) Рекурсивный метод часто более компактен
- 2) Итеративный способ менее требователен к памяти
- 3) Легкость проверки корректности значений в рекурсии
- 4) Однозначность назначения каждой переменной в рекурсии
- 5) Рекурсия выполняется определенное вложенное количество раз (как

правило, не больше 50 вызовов)

4. Определить основные особенности работы рекурсивных функций в языке программирования C++

В рекурсии обязательно должно быть возвращаемое значение, как и в любой not-void функции

Рекурсия имеет определенный предел вложенности (в зависимости от задачи, но как правило, это кол-во достигает ~50-ти)

Для выполнения рекурсивной функции происходит вызов стека памяти

5. Что такое прямая и непрямая (опосредкована) рекурсия?

Прямая рекурсия – функция вызывает сама себя определенное кол-во раз, пока не выполнится условие для ее завершения.

Косвенная рекурсия – функция А вызывает ф-ию В, которая вызывает дальше ф-ию А. И так далее, пока так же не выполнятся условия для прекращения рекурсий.

6. Когда выполняется завершение рекурсивных вызовов? Что такое рекурсивное заикливание и для чего оно приводит?

Завершение рекурсивного вызова происходит когда ф-ия достигла такого условия, когда ф-ия возвращает не саму себя, а какое-то значение.

Рекурсивное заикливание – когда не выполняется условие выхода из рекурсии и рекурсия начинает выполняться бесконечное количество раз, что приводит к огромным затратам (утечке) памяти и в дальнейшем – аварийному завершению программы.

7. Что такое область действия переменных?

Это специальная область какого-то объекта, в котором будет находиться какая-то переменная с определенным именем и значением.

Эта область нужна для того, что бы в разных местах программы использовать одинаковые имена переменных, но с разными значениями.

8. Как необходимо выполнить объявление переменной, что бы доступ к ней был возможен из любой ф-ии поточного файла программы?

Нужно объявить глобальную переменную, т.е. переменную, которая не будет



находится ни в одной из функций программы.

9. Как изменяются значения локальных и глобальных переменных при использовании рекурсивных ф-ий?

Локальная переменная, которая используется в рекурсии, после выхода из нее (рекурсии) удаляется из памяти, в отличие от глобальной, значение которой изменяется и сохраняется дальше.

10. Определите главные ограничения при использовании рекурсий.

- 1) Необходимость возвращать какое-то значение
- 2) Глубина рекурсии
- 3) Вызов стека памяти
- 4) Снижение производительности