Министерство цифрового развития, связи и массовых коммуникаций Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Санкт-петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича»

(СПбГУТ)

ОтчЁт  
ЛАБОРАТОРНАЯ РАБОТА №5

**Программирование вложенных циклов**

Руководитель,  
старший преподаватель Помогалова А. В.

подпись, дата

Исполнитель,  
группа ИКПИ-33 Коньков М. Д.

подпись, дата

УСЛОВИЯ РАБОТЫ

**Вариант № 12:**

Создать программу по вычислению значений функций y и z:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Номер | Функция | Рабочий набор | | | | |
| x | m | xнач | xкон | n |
| 12 |  | - | 10 | 3 | 8 | 15 |

**Общая формулировка:**

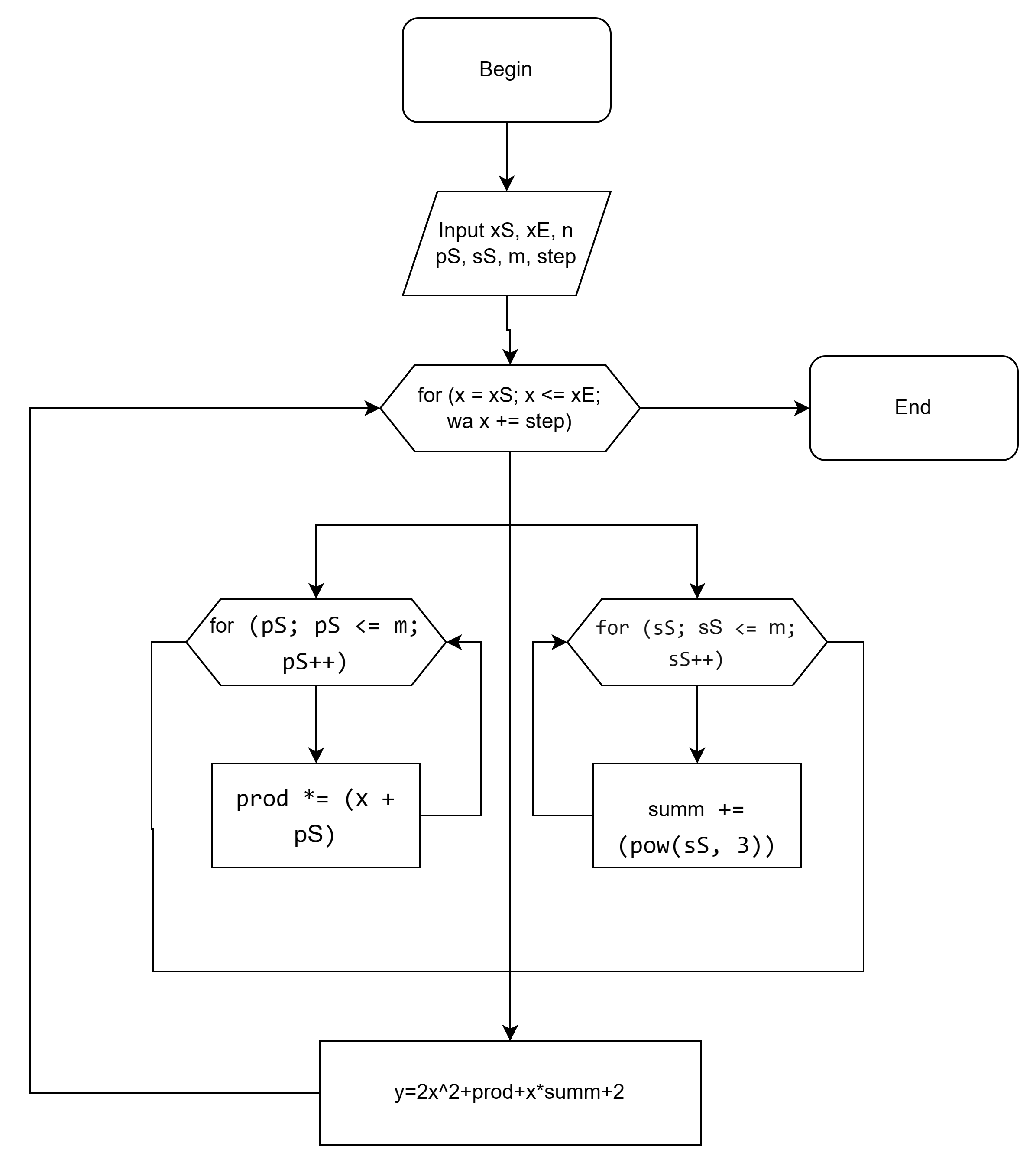
Необходимо выполнить вычисления, для организации которых следует использовать несколько циклов, причем некоторые из них должны быть вложенными.

# Общие алгоритм решения

Решаемая задача относится к категории задач формульного счета. Для нахождения переменной результата y необходимо пройти несколько итераций операции суммы и произведения и получить ответ. Это называется рекуррентной функцией. Потом нужно посчитать значение y аналогичным способом при помощи рекуррентной функции.

В программе должен быть предусмотрен ввод исходных данных, к которым относятся переменные *xS, xE, n, pS, sS, m*; вычисления величин *prod, summ, step*; вывод результатов вычислений (вывод значений величины *y*).

## Общий алгоритм решения

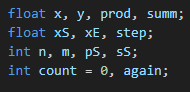


|  |  |  |  |
| --- | --- | --- | --- |
| N | Обозначение в задаче | Идентификатор | Назначение |
| 1 | *y* | *y* | Результат |
| 2 | *Xнач, Xкон, n, pS, sS, m, count* | *xS, xE, n, pS, sS, m, count* | Исходные данные |
| 3 | *∑, ∏, step* | *Sum, prod, step* | Промежуточная величина |

## Тестирование

Для тестирования программы выбираем контрольный набор исходных данных: *xS = 3, xE = 8, n = 15, а также pS = 1, sS = 1 и m = 10.*

**Переменные**:



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Переменная | xS | xE | n | pS | sS | m |
| Рабочий набор | 3 | 8 | 15 | 1 | 1 | 10 |
| Контрольный набор | 4 | 6 | 4 | 2 | 3 | 6 |

1. Листинг программы laba6.c (C)

#include <stdio.h>

#include <math.h>

int main(){

    float x, y, prod, summ;

    float xS, xE, step;

    int n, m, pS, sS;

    int count = 0, again;

    while(1){

        //// Entering and checking variables

        // xS, xE (entering and checking)

        printf("\033[0d\033[2J");

        printf("Enter x (start): \n");

        scanf("%f", &xS);

        printf("Enter x (ending): \n");

        scanf("%f", &xE);

        if (xS < 0 || xE < 0 || xS > xE)

        {

            while(xS < 0 || xE < 0 || xS > xE)

            {

                printf("Try again, values should be '> 0' and 'start < end' \n");

                printf("Enter x (start): \n");

                scanf("%f", &xS);

                printf("Enter x (ending): \n");

                scanf("%f", &xE);

                continue;

            }

        }

        printf("xS: %f; xE: %f \n", xS, xE);

        // n-step (entering and checking)

        printf("Enter step count: \n");

        scanf("%d", &n);

        if (n < 2)

        {

            while(n < 2)

            {

                printf("Try again, count of steps should be '>= 2' \n");

                printf("Enter step count: \n");

                scanf("%d", &n);

                continue;

            }

        }

        step = (xE - xS) / (float)(n);

        // pS, sS and m (entering and checking)

        printf("Enter prod start: ");

        scanf("%d", &pS);

        printf("Enter summ start: ");

        scanf("%d", &sS);

        printf("Enter the ending of summ and prod: ");

        scanf("%d", &m);

        if (pS < 1 || sS < 1 || m < 1)

        {

            while(pS < 1 || sS < 1 || m < 1)

            {

                printf("Try again, your values should be '> 0' \n");

                printf("Enter prod start: ");

                scanf("%d", &pS);

                printf("Enter summ start: ");

                scanf("%d", &sS);

                printf("Enter the ending of summ and prod: ");

                scanf("%d", &m);

                continue;

            }

        }

        if (pS >= m || sS >= m)

        {

            while(pS >= m || sS >= m){

                printf("Try again, your 'prod / summ' is 'bigger / equal' than 'end' or '<= 1'\n");

                printf("Enter prod start: ");

                scanf("%d", &pS);

                printf("Enter summ start: ");

                scanf("%d", &sS);

                printf("Enter the ending of summ and prod: ");

                scanf("%d", &m);

            }

        }

        printf("pS: %d; sS: %d; m: %d; xS: %f; xE: %f; step: %f\n", pS, sS, m, xS, xE, step);

        // Main cycle

        for (x = xS; x <= xE; x += step)

        {

            // Prod in-cycle

            for (pS; pS <= m; pS++)

            {

                prod \*= (x + pS);

            }

            // Summ in-cycle

            for (sS; sS <= m; sS++)

            {

                summ += (pow(sS, 3));

            }

            y = 2 \* pow(x, 2) + prod + x \* summ + 2;

            count++;

            printf("%d. y = %f\n", count, y);

        }

        // Checking for repeat

        printf("\nDo you want to try again? (1 - yes / 0 - no): ");

        scanf("%d", &again);

        if (again == 1)

        {

            continue;

        } else {

            return 0;

        }

    }

    return 0;

}

1. Результат работы программы laba6.c (С)

