

Часть 1. Программирование циклического процесса.

Типы циклов.

Цель работы

Изучить технологию программирования циклических процессов и типы циклов, решить практическую задачу, требующую применения данной технологии.

Задание

Рассмотреть решение предложенной задачи с использованием всех трех видов циклов. Отладить программу с наиболее рациональным вариантом цикла.

Обосновать выбор.

Определить количество чисел последовательности $(n-1)^2/n$, ($n=1,2,3,\dots,\infty$) попадающих в интервал $[h, m]$, где h, m - вещественные числа ($m > h$), вводимые с клавиатуры. Вывести на экран найденные числа и их количество.

Проект программы

Ни нижняя, ни верхняя границы перебора по n нам не известны. Значение, например, $(n-1)^2/n$ при $n = \text{trunc}(m) + 1$ может быть меньше m , то есть точно определить верхнюю границу перебора возможности нет, что исключает из рассмотрения цикл `for`, хотя решить задачу с ним возможно. Далее, хотя бы одну итерацию цикла мы обязаны провести, чтобы установить наличие или отсутствие элементов последовательности на заданном отрезке (последовательность возрастающая). Но первый член последовательности, очевидно, равен 0, поэтому счётчик можно инициализировать сразу равный 2, проверив первый на вхождение в отрезок. То есть рационально использовать цикл с предусловием – `while`.

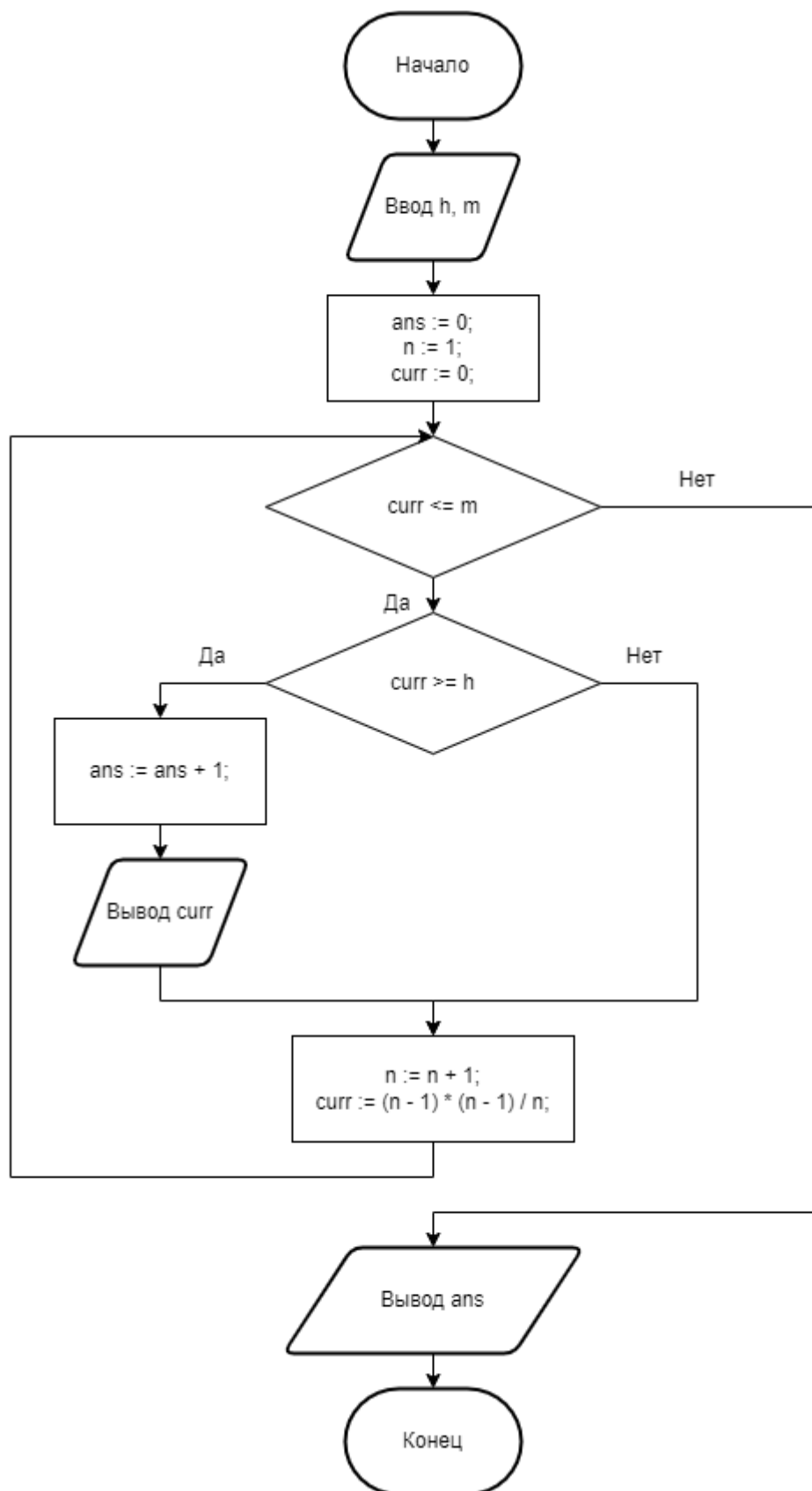
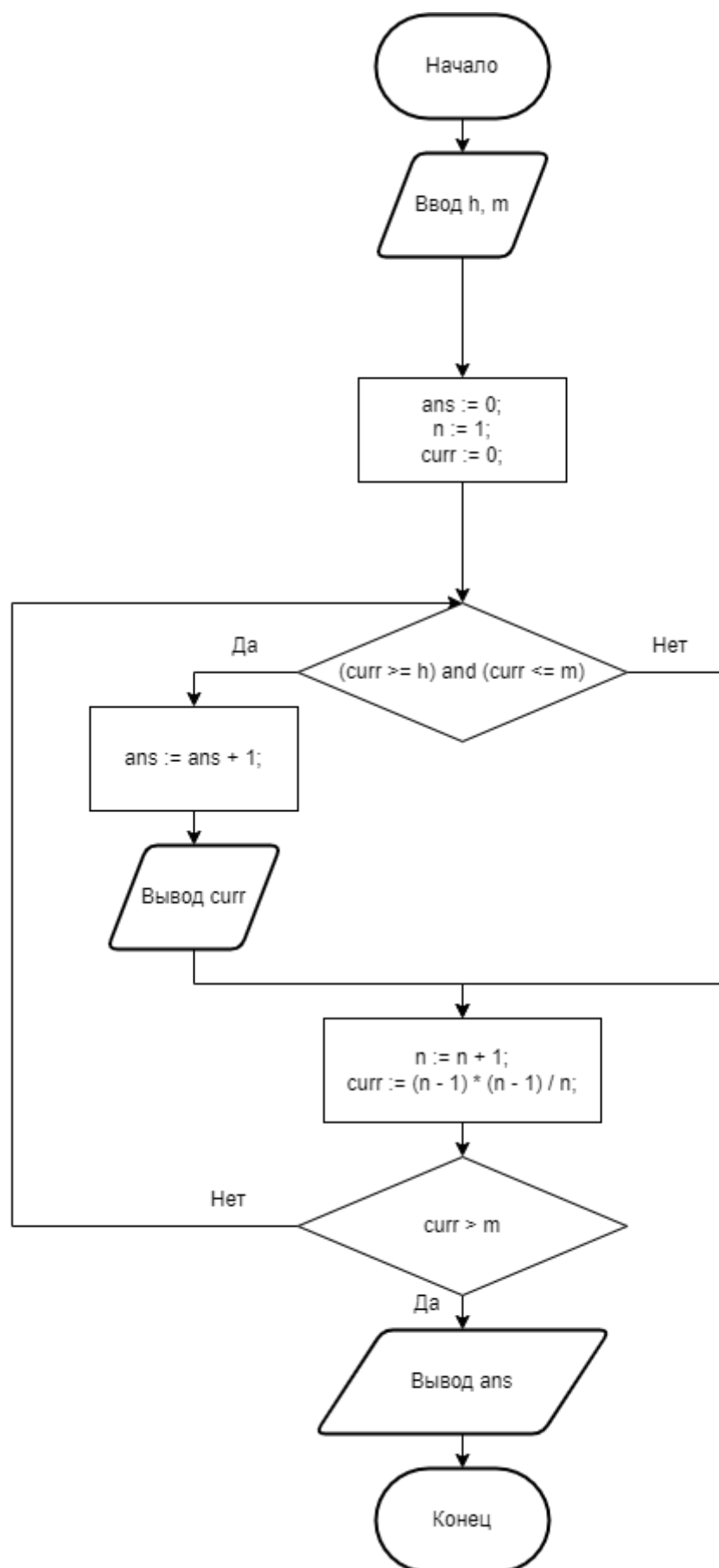
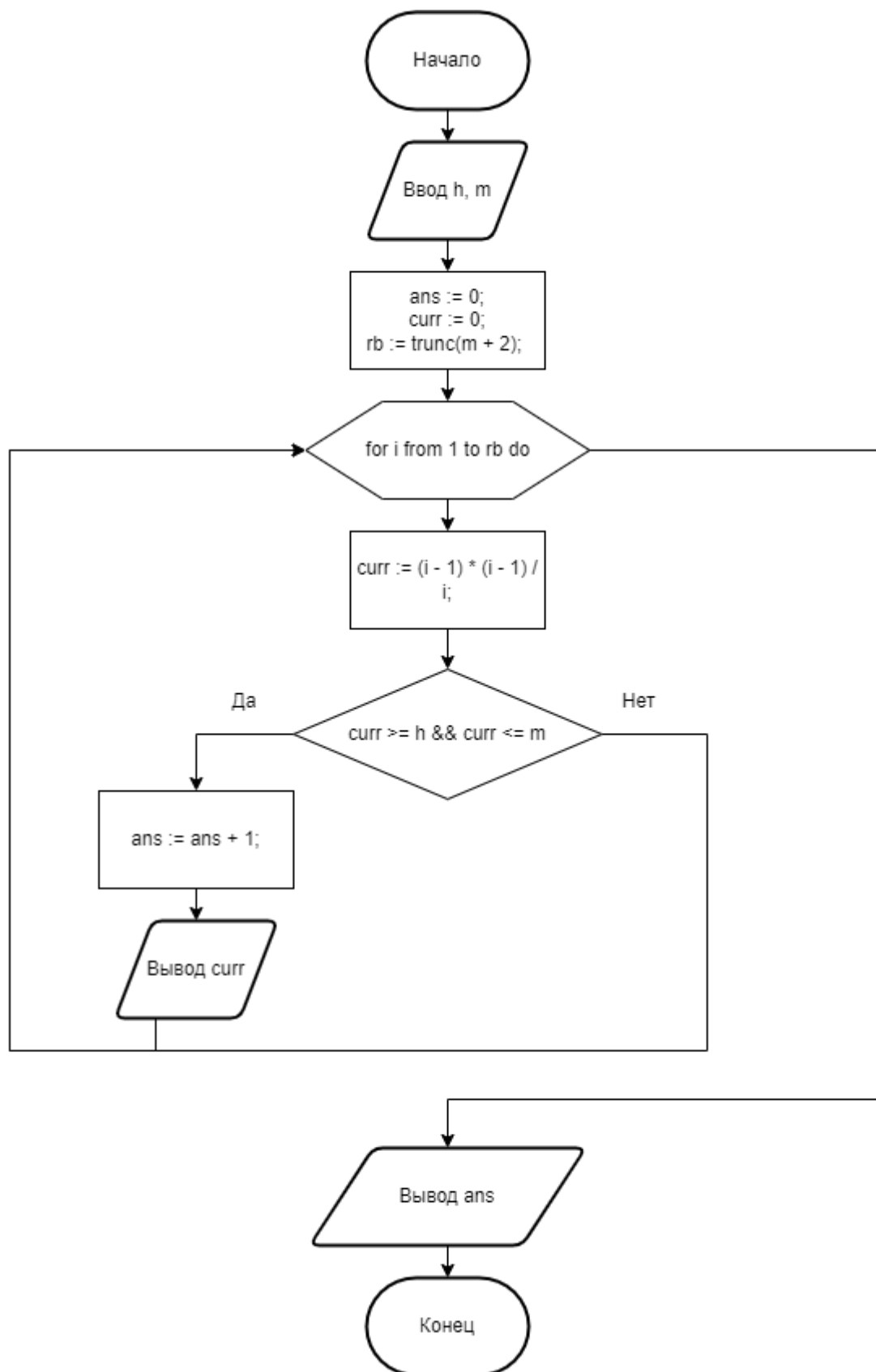


Рис. 1. Проект программы задания.





Текст программы

```
1  program n1;  
.  
.  
.  
.  
5  begin  
    writeln('Enter h, m: h < m');  
    readln(h, m);  
    curr := 0;  
    ans := 0;  
    n := 1;  
10 while curr <= m do  
    begin  
        if curr >= h then  
        begin  
            ans += 1;  
            writeln(curr);  
15         end;  
        n := n + 1;  
        curr := (n - 1) * (n - 1) / n;  
        end;  
20     writeln('Total numbers: ', ans);  
21     readln;  
.  
23 end.
```

Рис. 2. Текст программы задания while.

```
n2.lpr
1  program n2;
.
.  var h, m, curr: real; ans, n: integer;
.  begin
5    writeln('Enter h, m: h < m');
.    readln(h, m);
.    curr := 0;
.    ans := 0;
.    n := 0;
10   repeat
.     begin
.       n := n + 1;
13      curr := (n - 1) * (n - 1) / n;
.       if curr >= h then
15         begin
.           ans += 1;
.           writeln(curr);
.         end;
.       end
20   until curr > m;
.   writeln('Total numbers: ', ans);
.   readln;
.   end.
24
```

Рис. 3. Текст программы задания repeat..until.

```

.lpr
1  program n1;
.
.  var h, m, curr: real; ans, i, rb: integer;
.  begin
5    writeln('Enter h, m: h < m');
.    readln(h, m);
.    curr := 0;
.    ans := 0;
.    rb := trunc(m + 2);
10   for i := 1 to rb do
.     begin
.       curr := (i - 1) * (i - 1) / i;
13      if (curr >= h) and (curr <= m) then
.        begin
15          ans += 1;
.          writeln(curr);
.        end;
.      end;
.    writeln('Total numbers: ', ans);
20    readln;
.    end.
22

```

Рис. 4. Текст программы задания for.

Тестовые данные и результаты тестирования

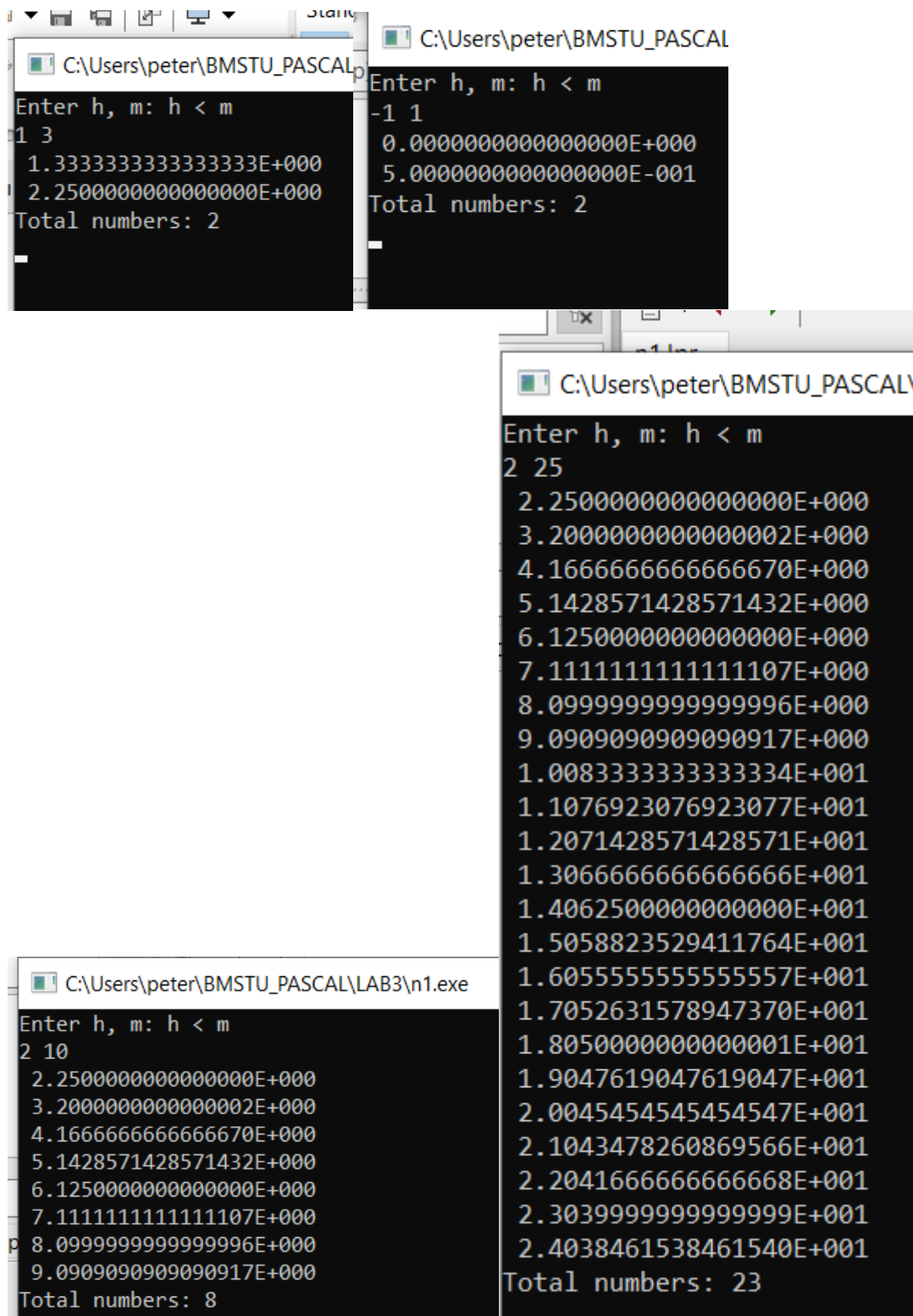


Рис. 5, 6, 7, 8. Результаты тестирования программы.

Вывод

Была изучена и применена при решении практической задачи технология программирования циклических процессов, были изучены типы циклов. Было установлено, что выбор типа цикла напрямую зависит от решаемой задачи.