**1.Таблица данных**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Описание  (смысл, диапазон, точность)** | **Тип** | **Структура** | **Формат в/в** |
| *Входные данные* | i | Номер варианта ответа  0<i<3 | цел | простая переменная | --- |
| n | Число, факториал которого нужно найти | вещ | простая переменная | XX.X (:4:1) |
| *Выходные данные* | otv | Найденный факториал заданного числа | вещ | простая переменная | XX.X (:4:1) |

Описание входных и выходных данных:

* i – входные данные, номер варианта ответа, целое число, простая переменная.
* n - входные данные, число, факториал которого нужно найти, вещественное число, простая переменная.
* otv – выходные данные, найденный факториал заданного числа, вещественное число, простая переменная.

программа написана на языке Delphi и реализована в виде одного файла, содержащего основной текст программы и подпрограмму, выполненную в виде функции языка Delphi. Основной текст программы можно разделить на три части:

* ввод исходных данных
* обработка данных
* вывод результата

**2.Пример работы программы**

Возьмем n=5.

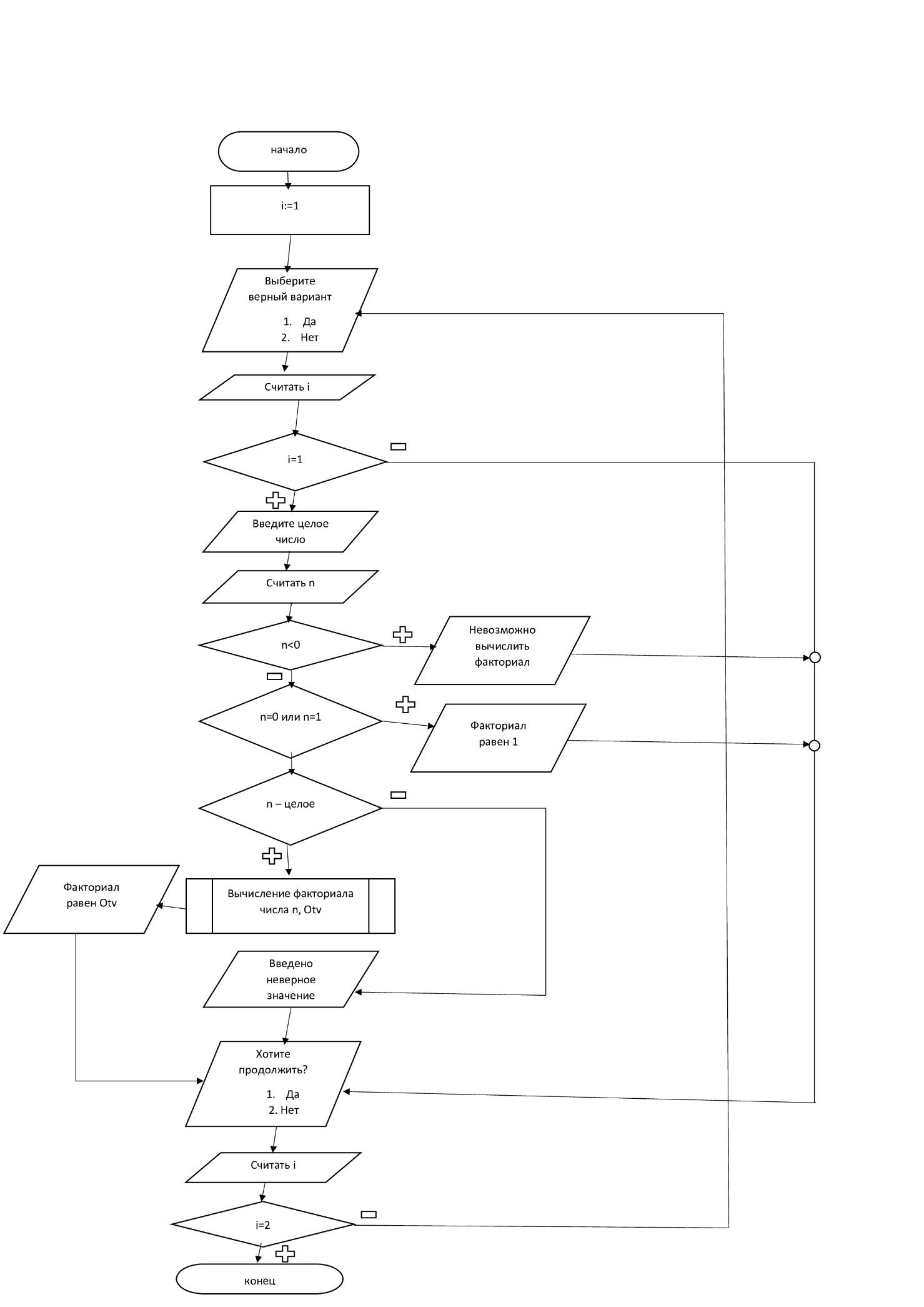
Можем заметить, что в каждом случае выражение вызывает функцию факториала (то есть саму себя) и уменьшает значение «n» на единицу, тем самым вычисляя факториал не с 1 , а с 5:

|  |  |  |
| --- | --- | --- |
| **Вызов функции** | **Вычисления** | **Описание** |
| Factorial:= 5 \* Factorial (5-1) | 5 \* Factorial (4) | Пока что мы не можем вычислить Factorial числа 5, так как мы вызвали функцию еще раз и необходимо вычислить теперь факториал 4 |
| Factorial:= 4 \* Factorial (4-1) | 4\* Factorial (3) | Пока что мы не можем вычислить Factorial числа 4, так как мы вызвали функцию еще раз и необходимо вычислить теперь факториал 3 |
| Factorial:= 3 \* Factorial (3-1) | 3\* Factorial (2) | Пока что мы не можем вычислить Factorial числа 3, так как мы вызвали функцию еще раз и необходимо вычислить теперь факториал 2 |
| Factorial:= 2 \* Factorial (2-1) | 2\* Factorial (1) | Пока что мы не можем вычислить Factorial числа 2, так как мы вызвали функцию еще раз и необходимо вычислить теперь факториал 1 |
| n = 1;  if n = 1 then Factorial:= 1 | Теперь n = 1 и функция прекращает своё действие и возвращаем 1 | Теперь программа можем вычислить факториалы поочередно от 1 до 5, так как мы больше не вызываем нашу функцию. |

Как только дошли до условия выхода из функции n = 1, получилось следующее:

|  |  |  |
| --- | --- | --- |
| **Вызов функции** | **Вычисления** | **Описание** |
| Factorial:= 2 \* Factorial (1) | 2\*1=2 | В Factorial (1) нам вернулось число 1 и записываем значение в Factorial = 2 |
| Factorial:= 3 \* Factorial (2) | 3\*2=6 | В Factorial (2) нам вернулось число 2 и записываем значение в Factorial = 6 |
| Factorial:= 4 \* Factorial (3) | 4\*6=24 | В Factorial (3) нам вернулось число 6 и записываем значение в Factorial = 24 |
| Factorial:= 5 \* Factorial (4) | 5\*24=120 | В Factorial (4) нам вернулось число 24 и записываем значение в Factorial = 120 |

Таким образом, факториал 5 равен 120.

**3.Описание логики работы (Блок-схема)**

**4. Аналитический анализ сложности**

**function** Faсtorial(n: real): real;

**begin**

**if** (n = 1) **then**

Faсtorial:= 1

**else**

**begin**

Faсtorial:= (n \* Faсtorial(n - 1));

**end**;

**end**;

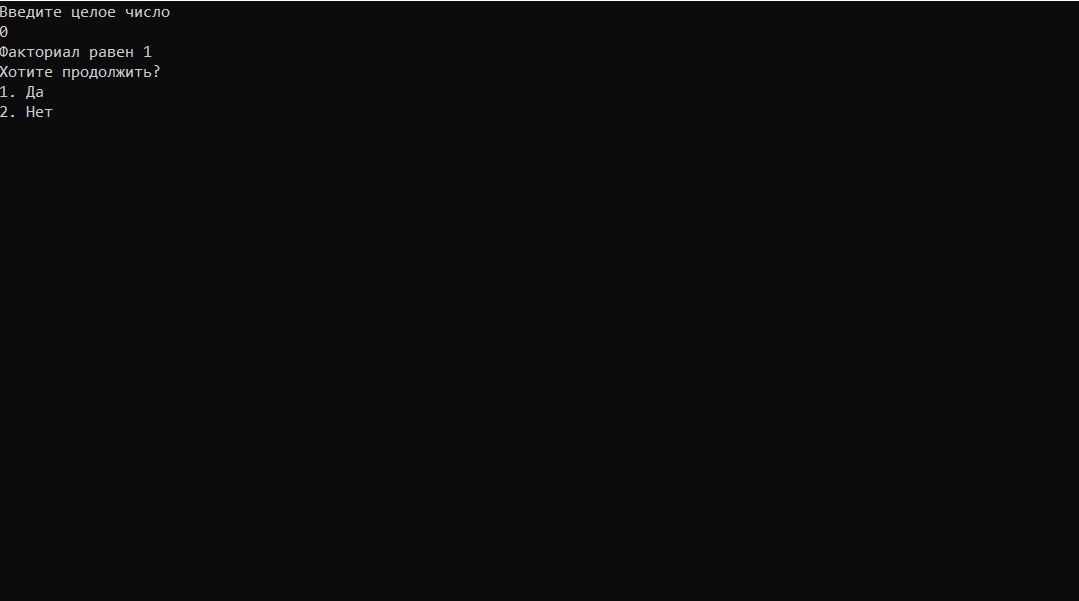
Необходимо заметить, что сложность алгоритма будет зависеть не только от сложности внутреннего цикла, но и от количества повторений.  
Эта процедура выполняется N раз, таким образом, вычислительная сложность этого алгоритма равна O(N).

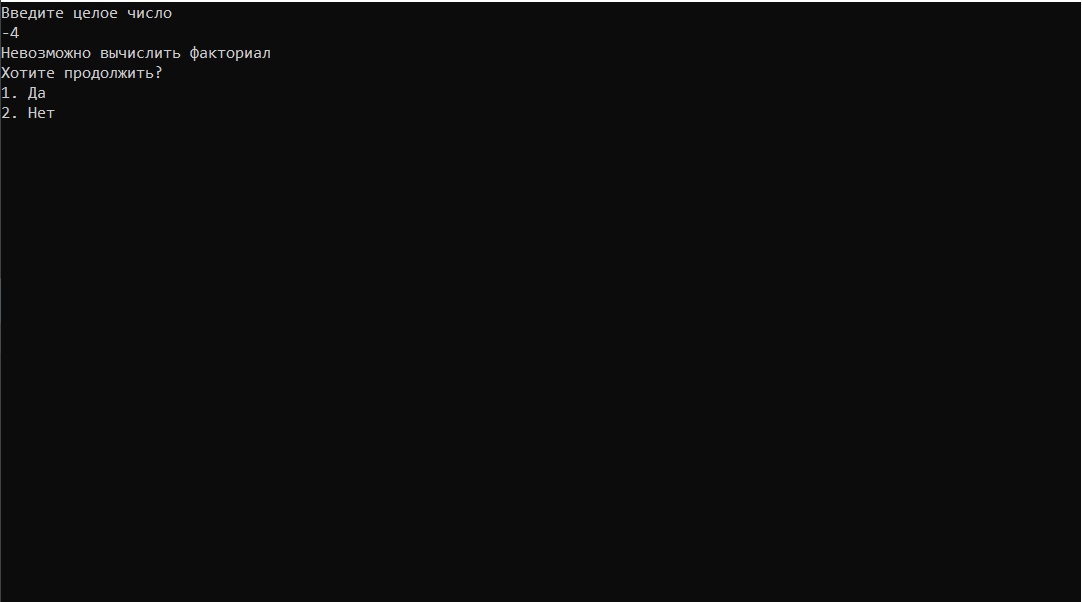
**5. Тесты**

|  |  |  |
| --- | --- | --- |
| **Входные данные** | **Ожидаемый результат** | **Смысл теста** |
| 0 | 1 | Минимальный вычисляемый факториал |
| -4 | Невозможно вычислить факториал | Аномальная ситуация -факториалов от отрицательных чисел не существует |
| 20 | 2432902008176640000 | Максимальный вычисляемый факториал |
| 1.5 | Введено неверное значение | Аномальная ситуация – программа не предполагает вычисление факториалов от дробных чисел |
| 9 | 362880 | Успешное вычисление факториала |

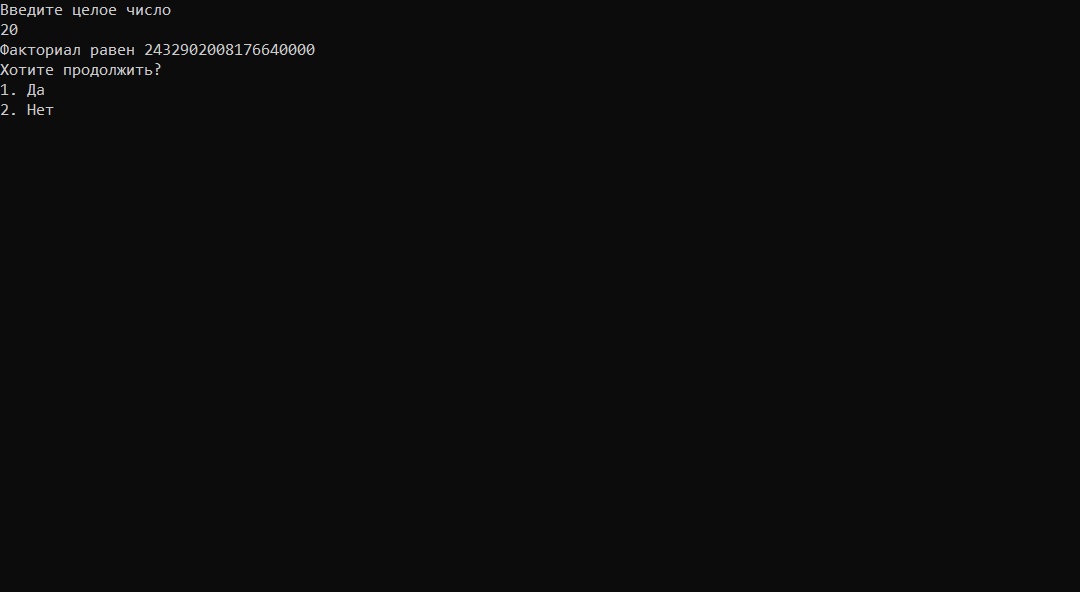
**Скриншоты тестов**

**Тест №1**

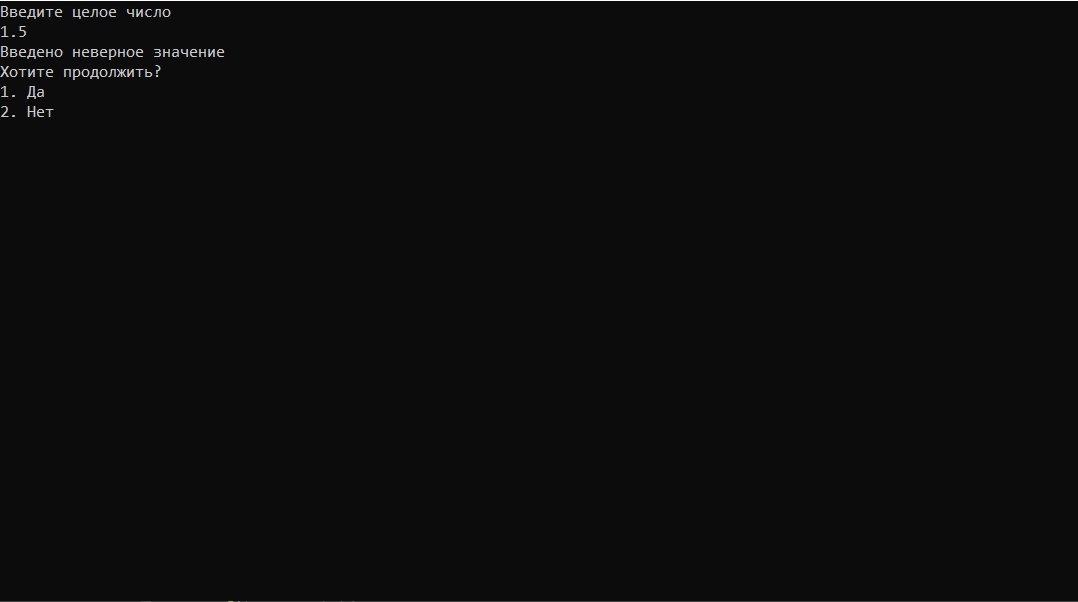


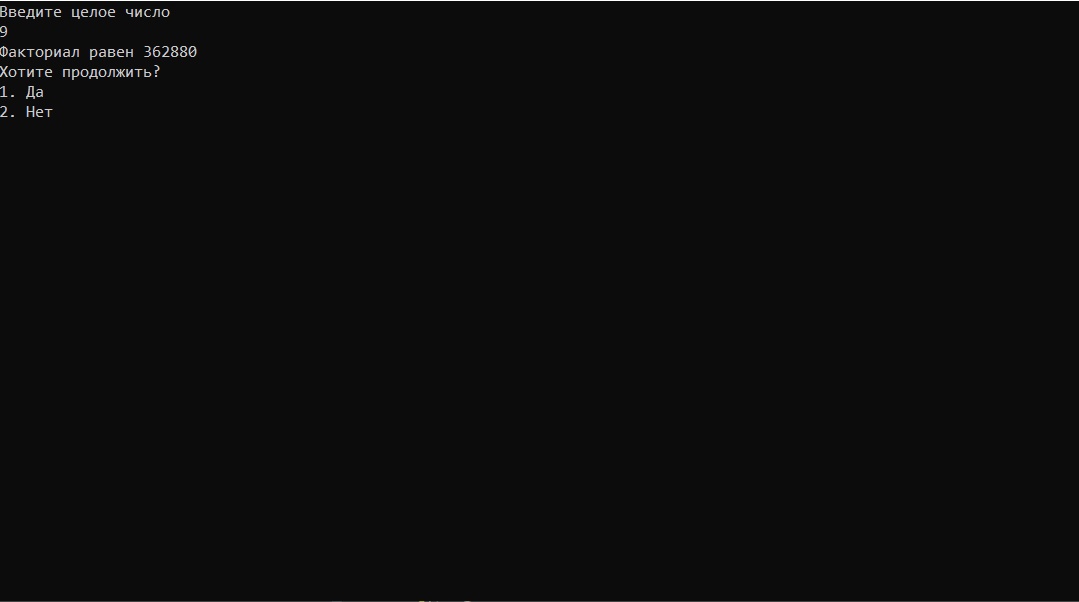
**Тест№2**

**Тест №3**



**Тест №4**



**Тест №5**

**5. Программный код**

**program** UD;

{$APPTYPE CONSOLE}

**function** Faсtorial(n: real): real;

**begin**

**if** (n = 1) **then**

Faсtorial:= 1

**else**

**begin**

Faсtorial:= (n \* Faсtorial(n - 1));

**end**;

**end**;

**var**

n,otv: real;

i: byte;

**begin**

i:= 1;

**repeat**

**if** (i <> 1) **then**

**begin**

writeln('Выберите верный вариант');

writeln('1. Да');

writeln('2. Нет');

readln(i);

**end**

**else**

**begin**

writeln('Введите целое число');

readln(n);

**if** (n < 0) **then**

writeln('Невозможно вычислить факториал')

**else**

**if** ((n = 0) **or** (n = 1)) **then**

writeln('Факториал равен 1')

**else**

**if** (trunc(n) = n) **then**

**begin**

otv:= Faсtorial(n);

writeln('Факториал равен ', otv)

**end**

**else**

writeln('Введено неверное значение');

writeln('Хотите продолжить?');

writeln('1. Да');

writeln('2. Нет');

readln(i);

**end**;

**until** (i = 2);

**end**.