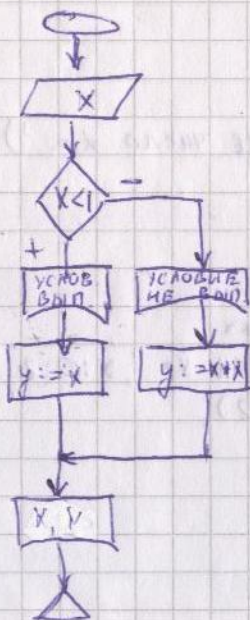


```

BEGIN
  CLRSCL;
  WRITE ('ВВЕДИТЕ ЧИСЛО X= ');
  READLN(X);
  IF X < 1
  THEN
    BEGIN
      Y := X;
      GOTO FLASH;
    END
  ELSE
    Y := X * X;
  FLASH: WRITELN ('X= ', X:0:2,
    ' TO Y= ', Y:0:2);
  READKEY;
END.

```

ПРИМЕР 2.2.4



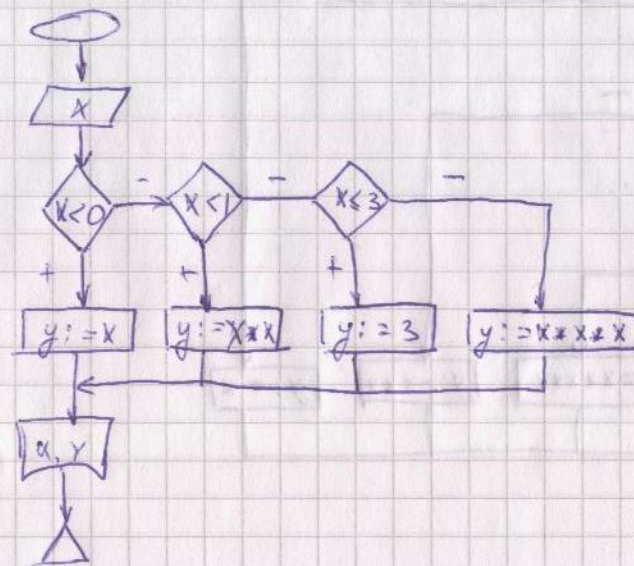
```

BEGIN
  CLRSCL;
  WRITE ('ВВЕДИТЕ ЧИСЛО X= ');
  READLN(X);
  IF X < 1
  THEN
    BEGIN
      WRITELN ('УСЛОВИЕ ВЫП. ');
      Y := X;
    END
  ELSE
    BEGIN
      WRITELN ('НЕ ВЫПОЛНЕНО'),
      Y := X * X;
    END;
  WRITELN ('X= ', X:0:2, ' Y= ', Y:0:2);
  READKEY;
END.

```

ПРИМЕР 2.2.5

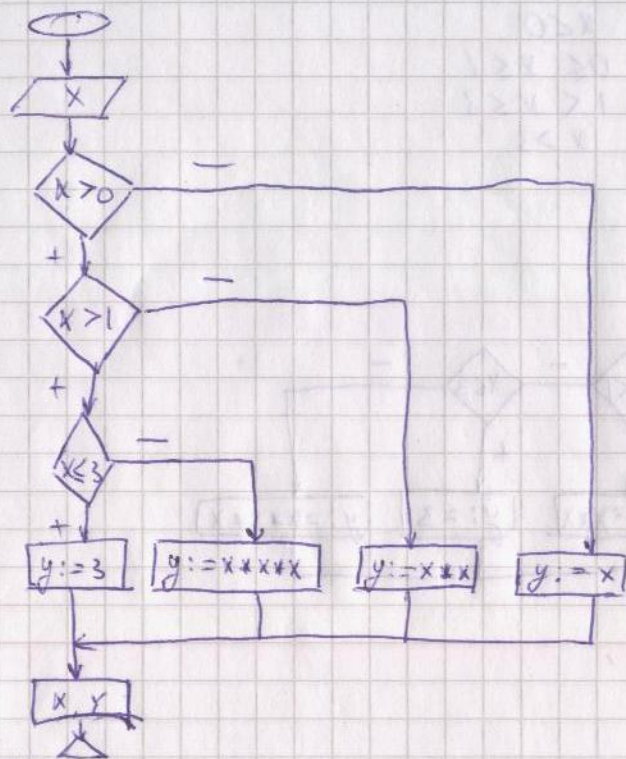
$$y = \begin{cases} x & \text{при } x < 0 \\ x^2 & \text{при } 0 \leq x \leq 1 \\ 3 & \text{при } 1 < x \leq 3 \\ x^3 & \text{при } x > 3 \end{cases}$$



```

BEGIN
  CLRSCL;
  WRITE ('ВВЕДИТЕ ЧИСЛО X= ');
  READLN(X);
  IF X < 0
  THEN
    Y := X;
  ELSE
    IF X <= 1
    THEN
      Y := X * X;
    ELSE
      IF X <= 3
      THEN
        Y := 3;
      ELSE
        Y := X * X * X;
      END;
    END;
  WRITELN ('X= ', X:0:2, ' Y= ', Y:0:2);
  READKEY;
END.

```

```

BEGIN
  CLRSCL;
  WRITE('ВВЕДИТЕ ЧИСЛО x = ');
  READLN(x);
  IF x > 0
  THEN
    IF x > 1
    THEN
      IF x <= 3
      THEN
        y := 3;
      ELSE
        y := x * x * x * x;
      ELSE
        y := x * x;
    ELSE
      y := x;
  WRITELN('x = ', x:0:2, ' y = ', y:0:2);
  READKEY;
END
  
```

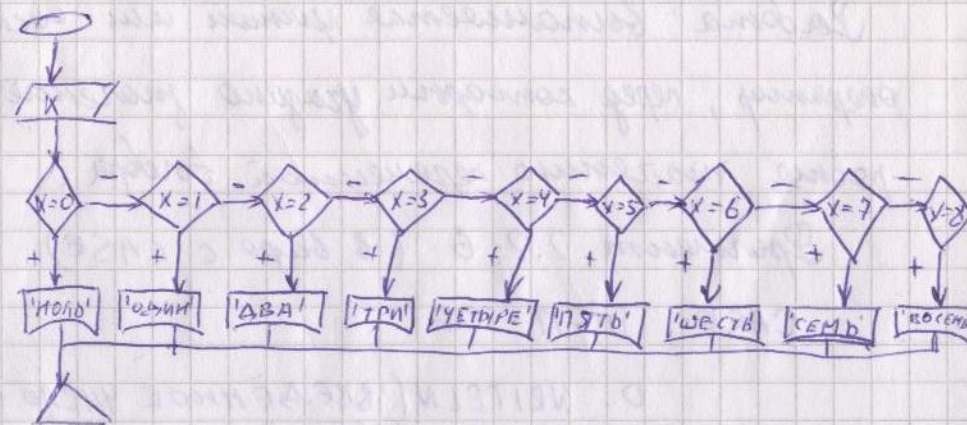
Вложенный оператор IF - оператор IF, в котором M или N является в свою очередь оператором IF.

IF A.B.1 THEN M1 ELSE IF A.B.2 THEN M2 ELSE N2, N1

IF A.B.1 THEN IF A.B.2 THEN M2 ELSE N2 ELSE N1

ПРИМЕР 2.2.6

Вводится число и его нужно вывести по-русски (текстом)



Когда условий разветвления много - программа громоздка, поэтому лучше использовать, если возможно, оператор варианта CASE (6 случаев).

Служит для осуществления разветвления
вычислительного процесса по ряду условий.

Формат: CASE переменная выбора OF
значение (значения): прямой или сост. оператор;

_____ " _____ : _____ " _____ ;
_____ " _____ : _____ " _____ ;

END

где CASE, OF, END - служебные слова;
переменная выбора - выражение любого типа,
кроме вещественного;

значение (одно или несколько) соответствующего типа.

Работа: выполняется прямой или составной
оператор, перед которым указано значение,
равное значению переменной выбора.

Фрагмент 2.2.6 (в виде с CASE)

CASE DIGIT OF

0: WRITELN('ВВЕДЁННОЕ ЧИСЛО - НУЛЬ');
1: WRITELN('ВВЕДЁННОЕ ЧИСЛО - ОДИН');
...

9: ...

END;

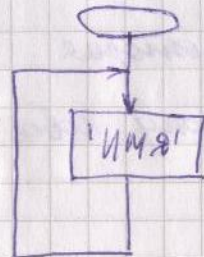
END.

CASE K OF

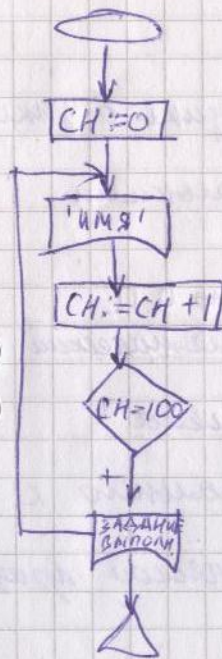
3, 6, 9 : WRITELN('...');

ELSE WRITELN('...')

2.3. Циклический вычислительный процесс



BEGIN
CLRSCR;
K: WRITE('ИМЯ');
GOTO K;
READKEY;
END.



BEGIN
CLRSCR;
CH:=0;
K: WRITE('ИМЯ');
CH:=CH+1;
IF CH=100
THEN
WRITELN('ЗАДАНИЕ ВЫПОЛН');
ELSE
GOTO K;
READKEY;
END.

Ц.в.п. - процесс, в котором вычисления
по одной и той же схеме алгоритма

выполняются многократно.

Циклическая программа - программа реализующая Ц.В.П.

Тело цикла - многократно повторяющийся фрагмент алгоритма

Параметр цикла - переменная, которая принимает по определённому закону новое значение при каждом новом цикле

Простой цикл - цикл с одним параметром цикла

Вложенный цикл (вложенный цикл, цикл в цикле) - циклическая программа с несколькими параметрами цикла

Арифметический цикл - циклическая программа, в которой параметр цикла меняет своё значение от начального к конечному по закону арифметической прогрессии

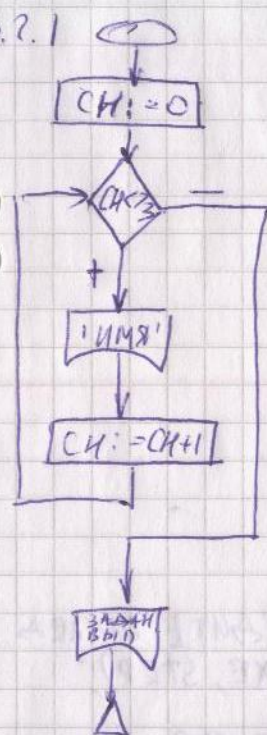
Итерационный цикл - циклическая программа, количество циклов которой

заранее неизвестно. И выход из цикла

выполняется посредством выполнения некоторого условия

2.3.2 Оператор цикла WHILE (цикл с предусловием)

2.3.2.1



```
BEGIN
  CLRSCR;
  CH := 0;
  S: IF CH < 3
  THEN
    BEGIN
      WRITELN('ИМЯ');
      CH := CH + 1;
      GOTO S;
    END
  ELSE
    WRITELN('ЗАДАНИЕ ВЫПОЛН');
  READKEY;
END.
```

Оператор WHILE служит для организации Ц.В.П.

Формат: WHILE Л.В. DO S,

WHILE, DO - служебные слова,

S - простой или составной оператор.

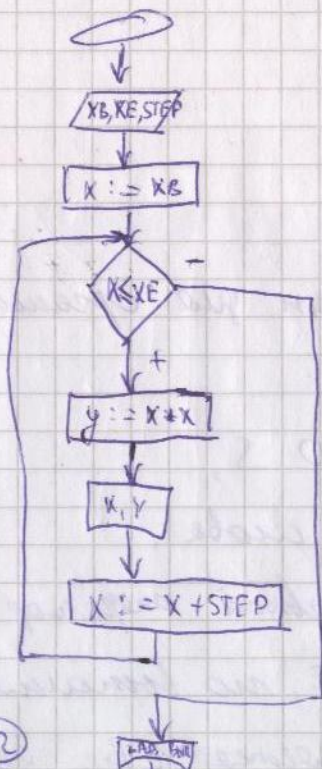
Работа: если Л.В. = TRUE, то выполняется S, затем вновь выполняется Л.В., если (31)

1. Б. = FALSE, то выполняется оператор
 до WHILE оператор.

```
BEGIN
  CH := 0;
  WHILE CH < 100 DO
    BEGIN
      WRITELN('...');
      CH := CH + 1;
    END;
END;
```

ПРИМЕР 2.3.2.2

$y = x^2$
 x = 2 y = 4
 x = 4 y = 16
 x = 6 y = 36



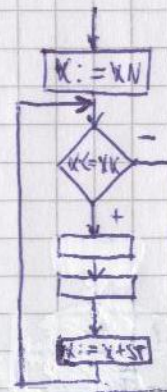
```
BEGIN
  CLRSCR;
  WRITELN('ВВЕДИТЕ ПОСЛЕД...');
  READLN(XB, KE, STEP);
  X := XB;
  WHILE X <= KE DO
    BEGIN
      Y := X * X;
      WRITELN('x = ', X, ' y = ', Y);
      X := X + STEP;
    END;
  WRITELN('ЗАДАНИЕ ВЫП...');
  READKEY;
END;
```

```
program alphabet;
uses
  CRT;
var
  BUKVA: CHAR;
```

```
BEGIN
  CLRSCR;
  WRITELN('ВВЕДИТЕ ЛЮБУЮ БУКВУ ');
  READLN(BUKVA);
  CASE BUKVA OF
    'А', 'а', 'Е', 'е', 'Ё', 'ё', 'И', 'и', 'О', 'о', 'У', 'у', 'Ы', 'ы',
    'Э', 'э', 'Ю', 'ю', 'Я', 'я': WRITELN('ВЫ ВВЕЛИ ГЛАСНУЮ');
  ELSE
    WRITELN('ВЫ ВВЕЛИ СОГЛАСНУЮ');
  END;
  READKEY;
END;
```

IF Л.Б. THEN ELSE } при пустом ELSE
 WRITELN('...'); } одиночном слитном
 } либо фактно присут-
 } ствовать

Алгоритм циклического
 возмущенного процесса
 с предсказанием



2.5.3 Оператор REPEAT (Ч.В.П. с ПОСТУПОВИЕМ)

REPEAT служит для реализации Ч.В.П.

Формат: REPEAT S UNTIL Л.В.

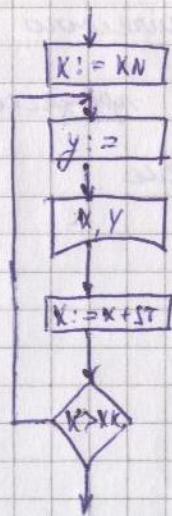
REPEAT
S
UNTIL Л.В.;

где REPEAT, UNTIL - служебные слова

S - один или несколько операторов

Работа: выполняется S, затем вычисляется Л.В., если Л.В. = FALSE - вновь выполняется S, если Л.В. = TRUE, то выполняется следующий оператор.

Алгоритм Ч.В.П. с поступовием



2.5.4 Оператор FOR

FOR служит для организации Ч.В.П.

Формат: FOR ПЧ:НЗПЧ TO(DOWNTO) КЗПЧ DO S

где FOR, TO(DOWNTO), DO - служебные слова

ПЧ (параметр цикла) - имя переменной любого типа, кроме REAL.

НЗ (начальное значение) и КЗ (конечное значение) - выражение соответствующего типа (по отношению к ПЧ)

S - простой или составной оператор

Работа: когда ПЧ - INTEGER при служебном слове TO, тогда при первом обращении к оператору принимает значение НЗПЧ, далее сравнивается с КЗПЧ и если ПЧ ≤ КЗПЧ, то выполняется S, после значение ПЧ увеличивается на единицу и цикл повторяется. Если же ПЧ > КЗПЧ, то выполняется следующий оператор.

BEGIN

FOR CH := 1 TO 10 DO
WRITELN('DYCH');

END.

Когда ПЧ - INTEGER при служебном слове

DOWNTO, тогда при первом обращении к оператору принимается значение КЗПЧ, далее сравнивается с КЗПЧ, если ПЧ $\geq$ КЗПЧ, то выполняется переход в тело цикла, далее значение ПЧ уменьшается на единицу и всё повторяется

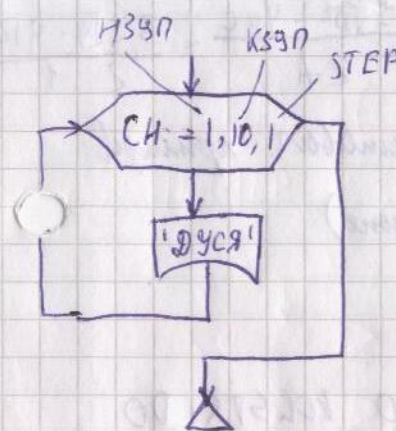
```
! BEGIN
  FOR LETTER := 'A' TO 'Z' DO
    WRITELN(LETTER, ' ');
  END.
```

Если ПЧ - переменная CHAR и содержит слово ТО, тогда только после S ПЧ принимает значение следующего символа.

```
BEGIN
  WRITELN('ПЕЧАТАЮ ВСЕ СИМВОЛЫ ПАСКАЛЯ');
  FOR SIM := CHR(0) TO CHR(255) DO
    WRITE(SIM, ' ');
  END.
```

Если ПЧ - BOOLEAN, то в качестве примера

```
PROGRAM NAME;
USES
  CRT;
VAR KEY: BOOLEAN;
BEGIN
  KEY := FALSE;
  FOR KEY TO TRUE DO
    WRITE(KEY);
  END.
```



Отчёт по лабораторным работам

Задача №

Фамилия, инициалы, группа

1. Формулировка задачи

Найти корни квадратного уравнения

$$Ax^2 + Bx + C = 0$$

2. Таблица соответствий

Переменная задачи	ЗАДАЧАТА
Имя в программе	Σ

3. Блок - схема алгоритма

4. Текст программы

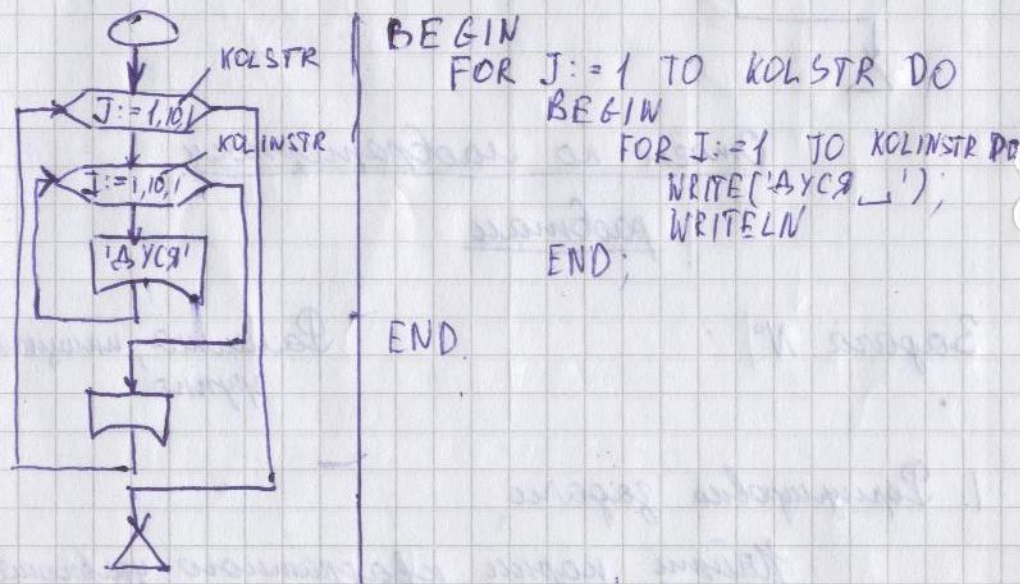
5. Тестовые примеры

$$(x-1)(x-2) = x^2 - 3x + 2 = 0$$

$$1) x^2 - 3x + 2 = 0; \quad x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} = \frac{3 \pm \sqrt{9 - 4 \cdot 1 \cdot 2}}{2 \cdot 1}$$

6. Результаты расчёта некоторых примеров на ЭВМ (все, как на экране)

ПРИМЕР 2.3.4.3



3. Пример разработки алгоритмов и программ минимальных вычислительных затрат

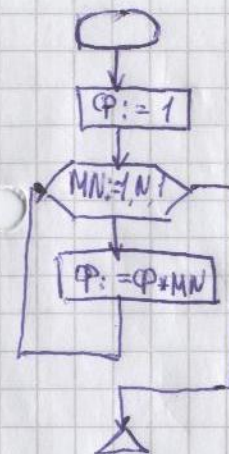
Пример 1. Вычислить $n!$

$$6! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \cdot 6 = 720$$

$$2! = 1 \cdot 2 = 2; \quad 1! = 1; \quad 0! = 1$$

$$MN := MN + 1$$

$$P := P \times MN$$

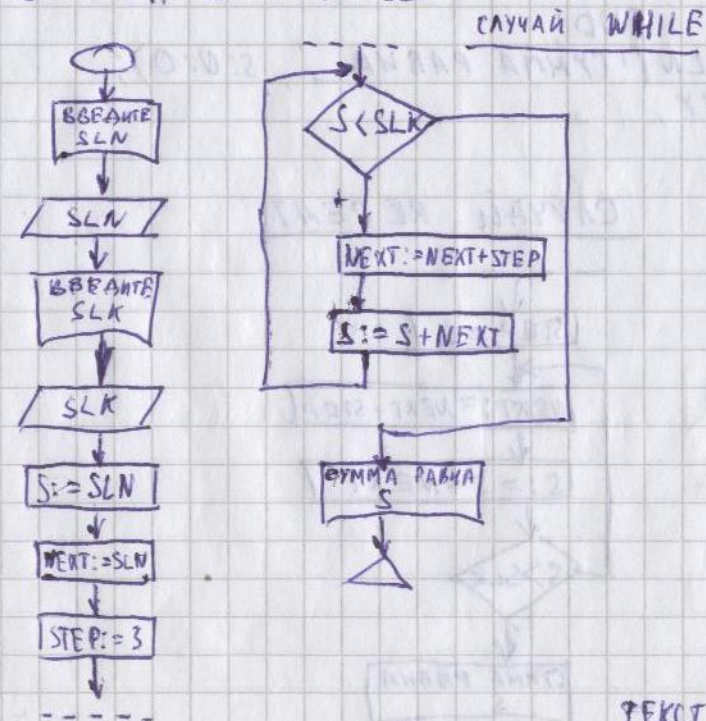


Пример 2

$$S = 15 + 18 + \dots + 123$$

$$3 \cdot 3 + 21 = 54 + 24 = 78 + 27 = 105 + 30 = 135$$

$$SUMMA := SUMMA + SL$$



ТЕКСТ ПРОГРАММЫ

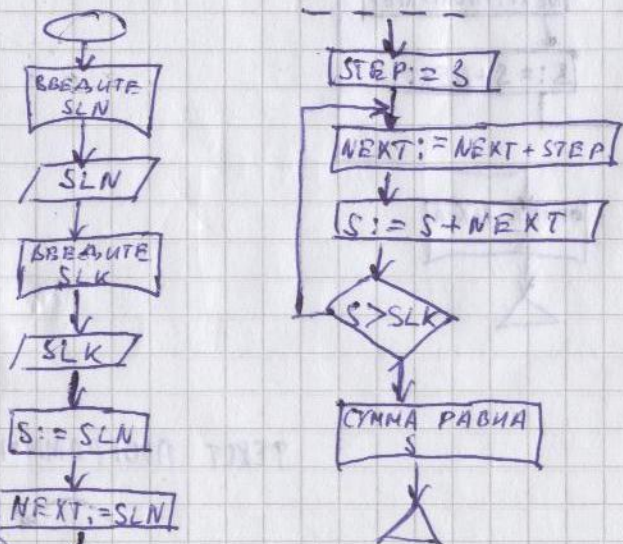

```

PROGRAM SUM1;
USES
  CRT;
VAR
  S, NEXT, STEP, SLN, SLK: REAL;

BEGIN
  CLRSCR;
  WRITE('ВВЕДИТЕ SLN');
  READLN(SLN);
  WRITE('ВВЕДИТЕ SLK');
  READLN(SLK);
  S := SLN;
  NEXT := SLN;
  STEP := 3;
  WHILE S < SLK DO
    BEGIN
      NEXT := NEXT + STEP;
      S := S + NEXT;
    END;
  WRITELN('СУММА РАВНА', S:0:0);
  READKEY;
END.

```

СЛУЧАЙ REPEAT



ТЕКСТ ПРОГРАММЫ

```

PROGRAM SUM2;
USES
  CRT;
VAR
  S, NEXT, STEP, SLN, SLK: REAL;

BEGIN
  CLRSCR;
  WRITE('ВВЕДИТЕ SLN');
  READLN(SLN);
  WRITE('ВВЕДИТЕ SLK');
  READLN(SLK);
  S := SLN;
  NEXT := SLN;
  STEP := 3;
  REPEAT
    NEXT := NEXT + STEP;
    S := S + NEXT;
  UNTIL S > SLK;
  WRITELN('СУММА РАВНА', S:0:0);
  READKEY;
END.

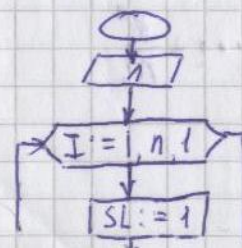
```

Отгөн по лабораторной работе (дополнение):

- 7. Результаты решения заданной задачи (где конкретные данные задачи)

Пример 3.

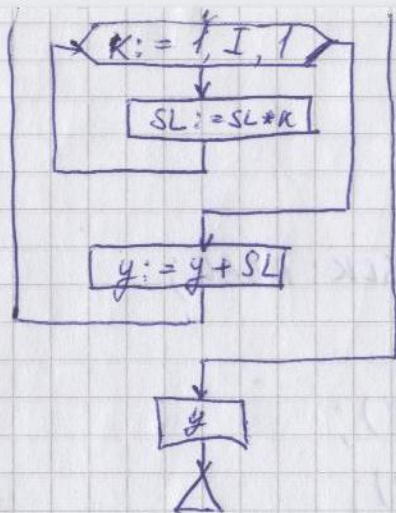
Вычислить $y = 1! + 2! + 3! + \dots + n!$ Дано n .



```

PROGRAM EXAMPLE 3;
USES
  CRT;
VAR
  SL, n, I, K, y: INTEGER;
BEGIN

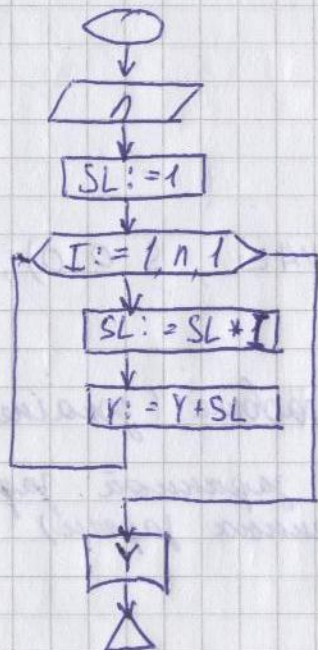
```

```

WRITE ('ВВЕДИТЕ ЧИСЛО N=');
READLN (n);
FOR I:=1 TO n DO
  BEGIN
    SL:=1;
    FOR K:=1 TO I DO
      SL:=SL*K;
    Y:=Y+SL
  END;
WRITELN ('ВЫЧИСЛЕНО ЗНАЧЕНИЕ ВЫРАЖЕНИЯ Y= ', Y);
READKEY;
END.

```



```

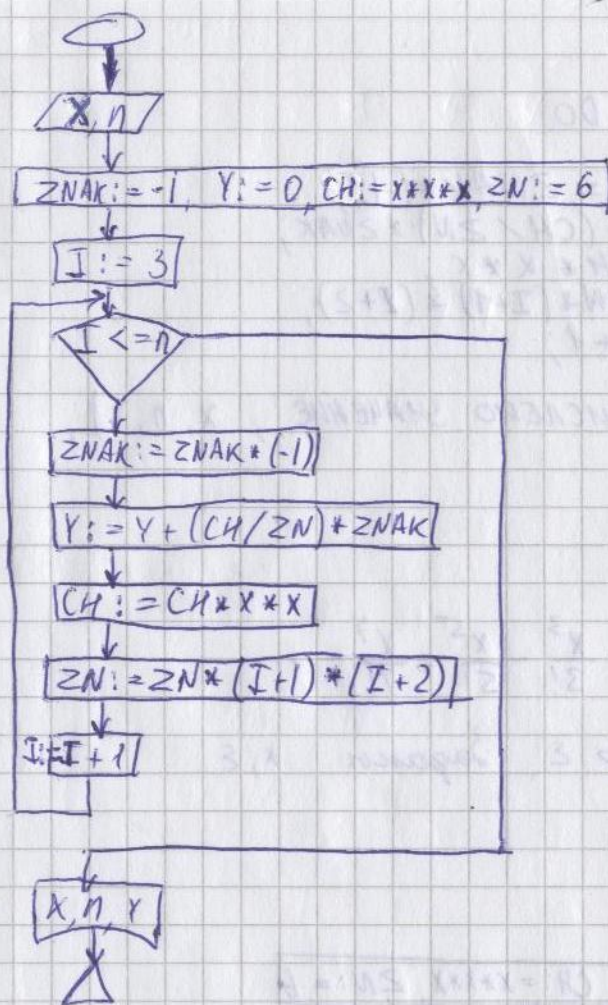
PROGRAM EXAMPLE 3_1;
USES
  CRT;
VAR
  SL, n, I, Y: INTEGER;
BEGIN
  WRITE ('ВВЕДИТЕ ЧИСЛО N=');
  READLN (n);
  SL:=1;
  FOR I:=1 TO n DO
    BEGIN
      SL:=SL*I;
      Y:=Y+SL;
    END;
  WRITELN ('ВЫЧИСЛЕНО ЗНАЧЕНИЕ ВЫРАЖЕНИЯ Y= ', Y);
  READKEY;
END.

```

ПРИМЕР 4

Вычислить $y = \frac{x^3}{3!} - \frac{x^5}{5!} + \frac{x^7}{7!} - \dots + \frac{x^n}{n!}$

Дано x, n .



ТЕКСТ ПРОГРАММЫ

```

PROGRAM EXAMPLE 4;
USES
  CRT;
VAR
  n, I, ZNAK: INTEGER;
  x, Y, CH, ZN: REAL;
BEGIN
  WRITE ('ВВЕДИТЕ ЧИСЛА x и n ');
  READLN (x, n);
  ZNAK := -1;
  Y := 0;
  CH := x*x*x;

```