



ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
УПРАВЛЕНИЕ ДИСТАНЦИОННОГО ОБУЧЕНИЯ И ПОВЫШЕНИЯ
КВАЛИФИКАЦИИ

Кафедра «Вычислительные системы и информационная
безопасность»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
по выполнению лабораторных работ
**«Алгоритмизация и системы
счисления»**
по дисциплине «Информатика»

Авторы
Смирнова О.В.
Полюян А.Ю.

Ростов-на-Дону, 2013



Аннотация

Методические указания предназначены для проведения лабораторных работ по дисциплине "Информатика" (для студентов первого курса всех специальностей и форм обучения). Лабораторные работы включают набор заданий, методические указания к ним и контрольные вопросы по изучаемой теме. Методические рекомендации могут быть использованы для самостоятельной работы.

Авторы

Кандидат технических наук, доцент Смирнова О.В.

Кандидат технических наук, доцент Полуян А.Ю.





Оглавление

ЛАБОРАТОРНАЯ	РАБОТА	№1
«АЛГОРИТМИЗАЦИЯ»		
5		
ТЕОРЕТИЧЕСКИЕ ОСНОВЫ АЛГОРИТМИЗАЦИИ		
5		
1.1	Понятие алгоритма. Свойства алгоритма	5
1.2	Формы записи алгоритмов	5
1.3	Алгоритмы линейной структуры	8
1.4	Алгоритмы разветвляющейся структуры	9
1.5	Алгоритмы циклической структуры	10
ВОПРОСЫ ДЛЯ САМОКОНТРОЛЯ		
16		
ИНДИВИДУАЛЬНЫЕ ЗАДАНИЯ.....		
17		
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ.....		
20		
ЛАБОРАТОРНАЯ	РАБОТА	№2
«ПРОГРАММИРОВАНИЕ		
АЛГОРИТМОВ		
РАЗВЕТВЛЯЮЩИХСЯ И ЦИКЛИЧЕСКИХ СТРУКТУР»		
.....		
21		
1. ЦЕЛЬ РАБОТЫ		
21		
2. ПОРЯДОК ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ		
21		
3. КРАТКИЕ СВЕДЕНИЯ ИЗ ТЕОРИИ.....		
21		
3.1.	Программирование алгоритмов разветвляющейся структуры	22
3.2.	Программирование алгоритмов циклической структуры	26
3.3	Программирование алгоритмов со структурой вложенных циклов.....	31
4. КОНТРОЛЬНЫЕ ВОПРОСЫ.....		
33		
5. ИНДИВИДУАЛЬНЫЕ ЗАДАНИЯ.....		
34		
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ.....		
38		



ЛАБОРАТОРНАЯ РАБОТА №3	«СИСТЕМЫ
СЧИСЛЕНИЯ»	39
1. ЦЕЛИ РАБОТЫ	39
2. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ	40
3. ЗАДАНИЯ ДЛЯ ВЫПОЛНЕНИЯ САМОСТОЯТЕЛЬНОЙ	
РАБОТЫ	48
ВОПРОСЫ ДЛЯ САМОКОНТРОЛЯ	50
ЛИТЕРАТУРА	51



ЛАБОРАТОРНАЯ РАБОТА №1

«АЛГОРИТМИЗАЦИЯ»

Содержит общие сведения об алгоритмах, позволяет освоить: способы описания алгоритмов, основные приемы построения алгоритмов для решения различных задач.

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ АЛГОРИТМИЗАЦИИ

1.1 Понятие алгоритма. Свойства алгоритма

Алгоритм - это последовательность действий, приводящих к требуемому результату.

Таким образом, при разработке алгоритма решения задачи математическая формулировка преобразуется в процедуру решения, представляющую собой последовательность арифметических действий и логических связей между ними. При этом алгоритм обладает следующими свойствами:

- 1) Дискретность - процесс преобразования данных, т.е. на каждом шаге алгоритма выполняется очередная одна операция;
- 2) Результативность - алгоритм должен давать некоторый результат;
- 3) Конечность - алгоритм должен давать результат за конечное число шагов;
- 4) Определенность - все предписания алгоритма должны быть однозначны, понятны пользователю;
- 5) Массовость - алгоритм должен давать решения для целой группы задач из некоторого класса, отличающихся исходными данными;

Действия в алгоритме выполняются в порядке их записи. Нельзя менять местами никакие два действия алгоритма, а так же нельзя не закончив одного действия переходить к следующему.

1.2 Формы записи алгоритмов

На практике наиболее распространены следующие формы представления

алгоритмов:

- **словесная** (запись на естественном языке);
- **графическая** (изображения из графических символов);
- **псевдокоды** (полуформализованные описания алгоритмов

на услов-

ном алгоритмическом языке, включающие в себя как эле-



менты языка

программирования, так и фразы естественного языка, общепринятые



математические обозначения и др.);

- **программная** (тексты на языках программирования).

Словесный способ записи алгоритмов представляет собой описание последовательных этапов обработки данных. Алгоритм задается в произвольном изложении на естественном языке. Словесный способ не имеет широкого распространения, так как такие описания:

- строго не формализуемы;
- страдают многословностью записей;
- допускают неоднозначность толкования отдельных предписаний.

Графический способ представления алгоритмов является более компактным и наглядным по сравнению со словесным. При графическом представлении алгоритм изображается в виде последовательности связанных между собой функциональных блоков, каждый из которых соответствует выполнению одного или нескольких действий. Такое графическое представление называется схемой алгоритма. В схеме алгоритма каждому типу действий (вводу исходных данных, вычислению значений выражений, проверке условий, управлению повторением действий, окончанию обработки и т.п.) соответствует геометрическая фигура, представленная в виде блочного символа. Блочные символы соединяются линиями переходов, определяющими очередность выполнения действий. В таблице приведены наиболее часто употребляемые символы.

Название символа	Обозначение	Выполняемая функция
Начало/конец		Начало или конец алгоритма
Процесс вычисления		Выполняет вычислительное действие или группу действий



Логический блок		Выбор направления выполнения алгоритма в зависимости от условия
Ввод /вывод		Отображение данных
Граница цикла		Отображает начало и конец цикла
Предопределенный процесс		Выполнение операций в подпрограмме
Соединитель		Указание связи между прерванными линиями в пределах одной страницы
Комментарий		Пояснительная запись

Схема алгоритма выстраивается в одном направлении: либо сверху вниз, либо слева направо. Все повороты соединительных линий выполняются под углом 90 градусов.

Общими правилами при построении схем алгоритмов являются следующие:

- В начале алгоритма должны быть блоки ввода значений входных данных.
- После ввода значений входных данных могут сле-



довать процесс вычислений и блоки условия.

- В конце алгоритма должны располагаться блоки вывода значений выходных данных.
- В алгоритме должен быть только один блок начала и один блок окончания.

Связи между блоками указываются направленными или ненаправленными линиями.

Псевдокод представляет собой систему обозначений и правил, предназначенную для единообразной записи алгоритмов. Псевдокод занимает промежуточное место между естественным и формальным языками. С одной стороны, он близок к обычному естественному языку, поэтому алгоритмы могут на нем записываться и читаться как обычный текст. С другой стороны, в псевдокоде используются некоторые формальные конструкции и математическая символика, что приближает запись алгоритма к общепринятой математической записи.

1.3 Алгоритмы линейной структуры

Алгоритм, в котором действия выполняются последовательно друг за другом, называется **линейным алгоритмом**.

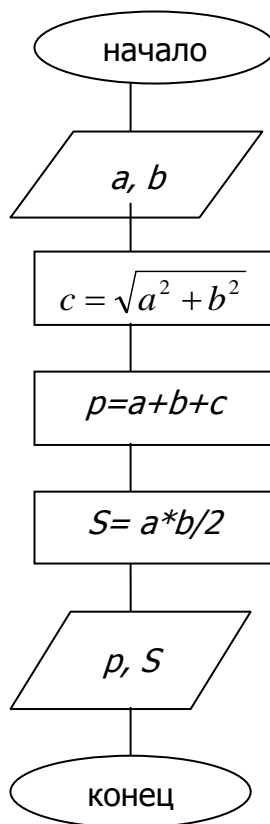
Рассмотрим пример составления схемы линейного алгоритма:



Пример 1. Вычислить периметр и площадь прямоугольного треугольника по длинам двух катетов.

Решение. Гипотенуза прямоугольного треугольника вычисляется по формуле Пифагора $c = \sqrt{a^2 + b^2}$, где a , b – катеты. Периметр и площадь прямоугольного треугольника определяются по формулам: $p = a + b + c$, $S = a \cdot b / 2$.

Схема алгоритма примера (рис.1) иллюстрирует, что сначала вводятся исходные данные a , b затем вычисляется гипотенуза, периметр, площадь. Все вычисления производятся последовательно и менять их местами нельзя.



1.4 Алгоритмы разветвляющейся структуры

На практике редко удастся представить схему алгоритма решения задачи

в виде линейной структуры. В программу может быть включено условие (на-

пример, выражение отношения или логическое отношение), в зависимости от

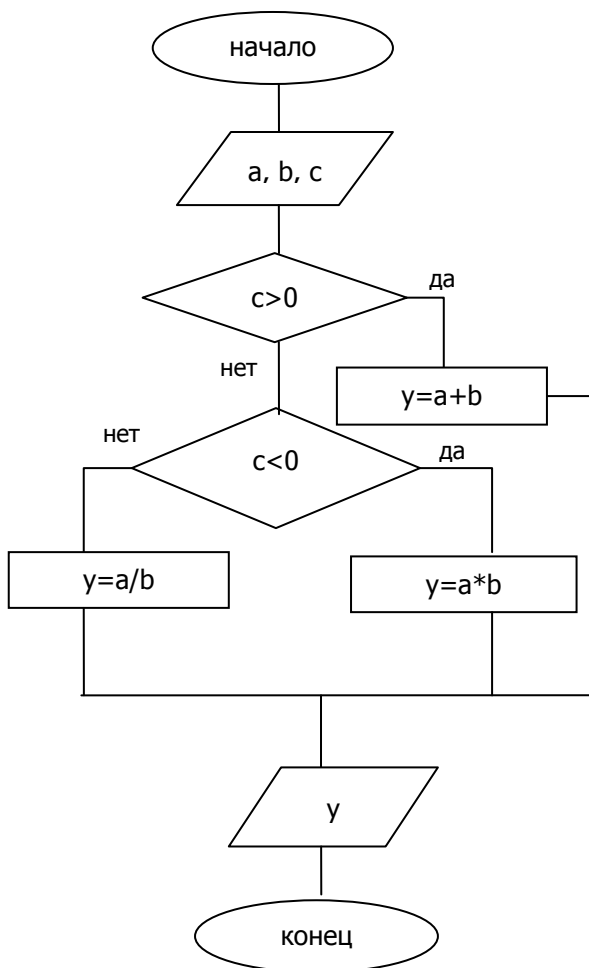
которого, вычислительный процесс идет по той или иной ветви. Алгоритм такого вычислительного процесса называется **алгоритмом разветвляющейся структуры**. В общем случае количество ветвей в таком алгоритме не обязательно равно двум.



Пример 2. Составить схему алгоритма вычисления выражения

$$y = \begin{cases} a + b, & c > 0 \\ a * b, & c < 0 \\ a / c, & c = 0 \end{cases}$$

После ввода исходных данных (переменных a , b , c) проверяется значение переменной c в логическом блоке. Если условие выполняется (истина), то после выполнения блоков первой ветви нет необходимости выполнять блоки второй ветви, осуществляется переход сразу к выводу результата и концу алгоритма. Если условие не выполняется (ложь), то переходим к проверке следующего логического условия. Решение указывается условием перехода на соответствующую ветвь.



1.5 Алгоритмы циклической структуры

Алгоритм, в котором вычисления повторяются по одной и той же совокупности формул, называется **циклическим алгоритмом**. *Цикл* – это многократно повторяемый участок алгоритма.

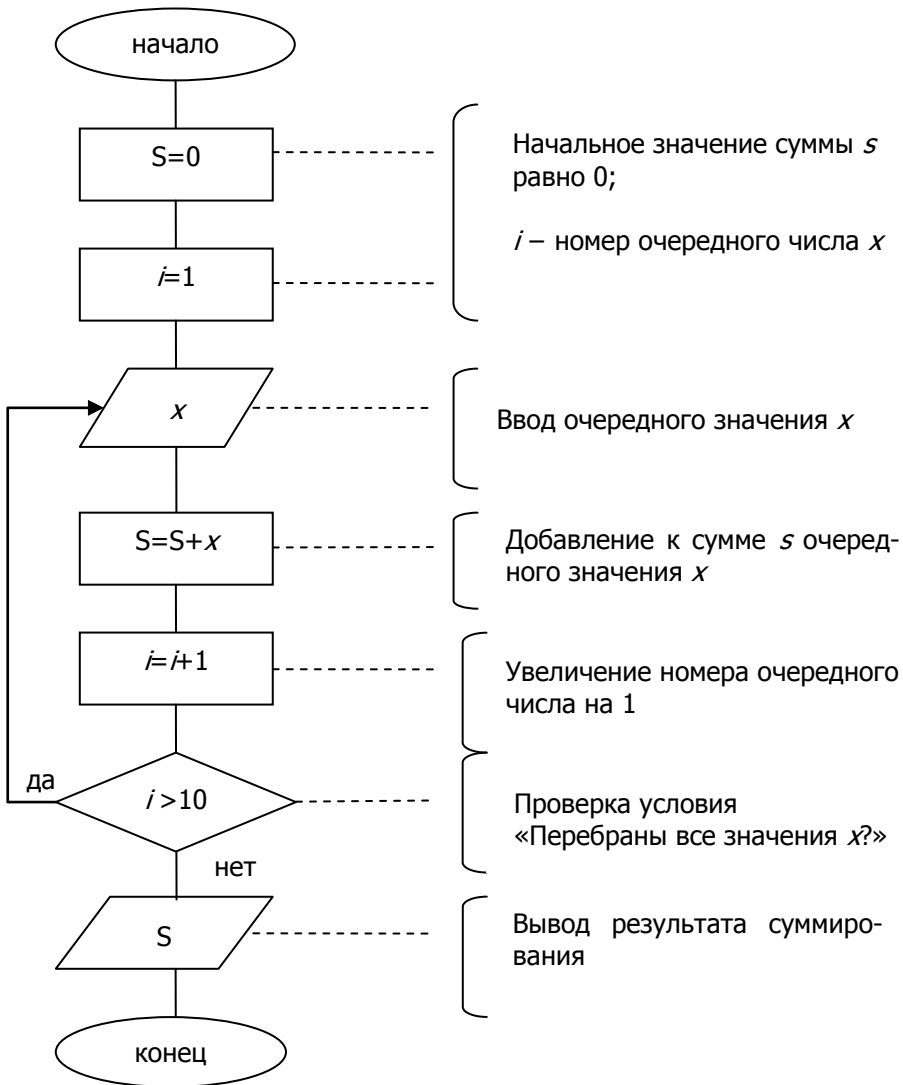
Массив – это совокупность переменных, которые имеют одно и то же имя и тип. Элементы массива различаются по индексу. Имя общее, индекс оригинальный. Упорядоченность данных в



массиве позволяет обращаться к любому элементу массива по его номеру (индексу), а однотипность данных позволяет использовать циклическую обработку всех элементов. Различают одномерные массивы (1 индекс) – они используются для представления векторов и двумерные массивы (2 индекса) они используются для представления матриц.

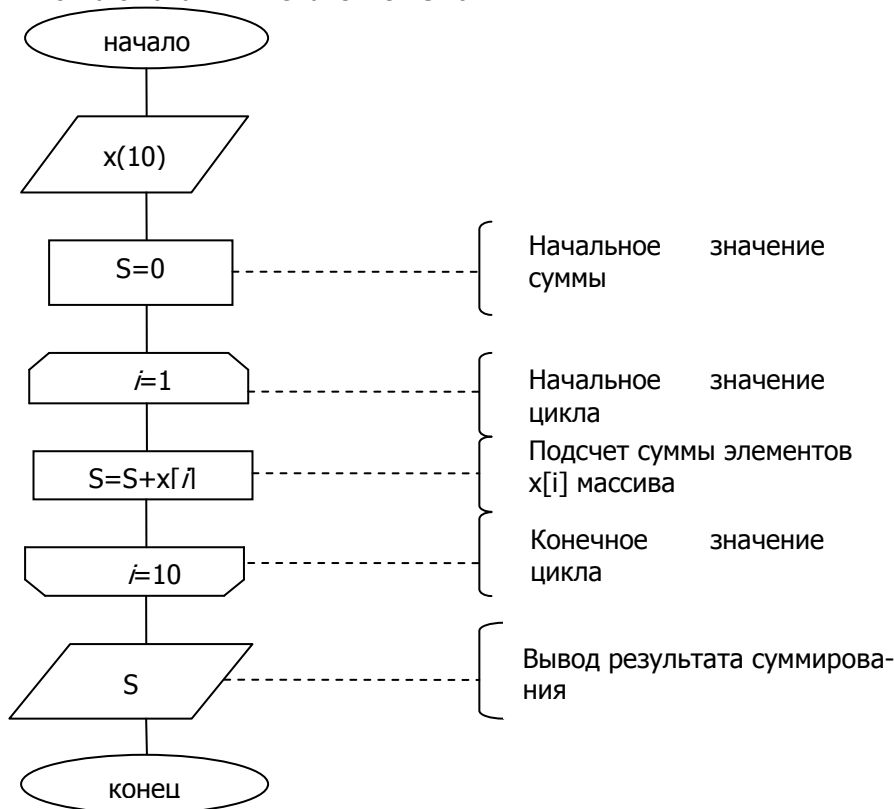
Пример 3. Составить циклический алгоритм вычисления суммы десяти чисел $S = \sum_{i=1}^{10} x_i$.

Вариант 1 построения алгоритма. Здесь в качестве переменной цикла используется переменная i с начальным значением, равным единице, и конечным значением, равным 10, и шагом, равным единице. В этом цикле проверка условия выхода из цикла выполняется в конце цикла. При этом тело цикла повторится десять раз.



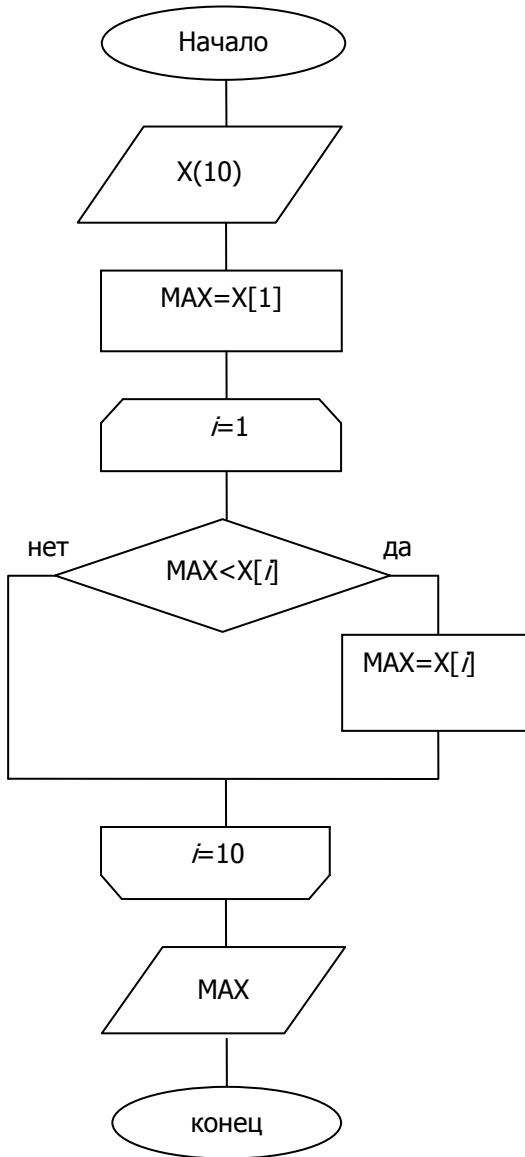


Вариант 2 построения алгоритма. Схема алгоритма получается во многих случаях более компактной и наглядной, если для ее построения использовать блоки начала и конца цикла, который выполняет все функции, необходимые для его организации. В цикле последовательно суммируются все элементы x_i массива с начальным значением $S=0$.



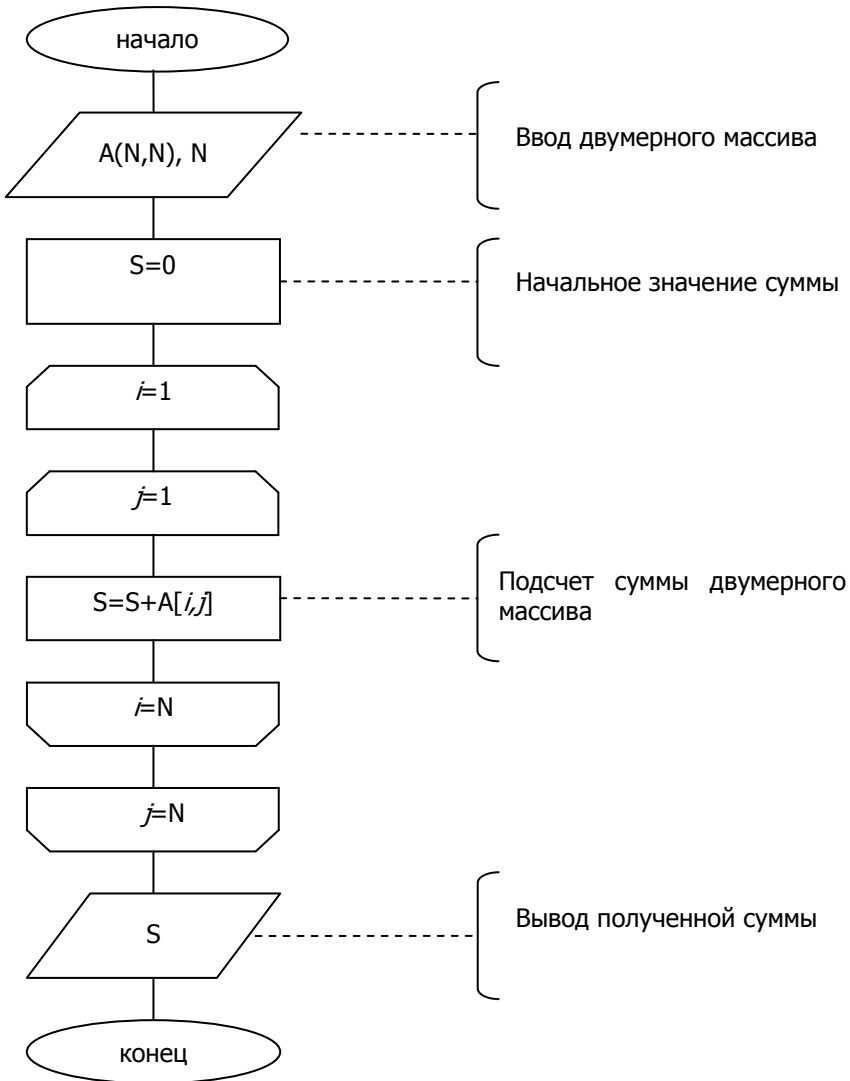
Пример 4. Найти наибольший элемент одномерного массива $X(10)$.

Процесс определения наибольшего элемента заключается в цикле сравнивая текущий элемент массива с некоторым, например, с первым, условно принятым за наибольшим. Если текущее значение массива окажется больше наибольшего из предыдущих значений, то его надо считать новым наибольшим значением.





Пример 5. Задан двумерный массив $A(N,N)$, найти сумму элементов массивов.





ВОПРОСЫ ДЛЯ САМОКОНТРОЛЯ

1. Понятие алгоритма
2. Свойства алгоритмов
3. Способы записи алгоритмов
4. Формы представления алгоритмов
5. Алгоритмизация линейных вычислительных процессов
6. Алгоритмизация ветвящихся вычислительных процессов
7. Алгоритмизация циклических процессов



ИНДИВИДУАЛЬНЫЕ ЗАДАНИЯ

1. Построить схему алгоритма $y = \sqrt{a^2 + b^2}$ где

$$a = \begin{cases} \operatorname{tg} x, & \text{если } x > 2 \\ x^2, & \text{если } 0 \leq x \leq 2 \\ 4, & \text{если } -1 \leq x < 0 \\ \sqrt{\sin x}, & \text{в остальных случаях} \end{cases}$$

2. Построить схему алгоритма

$$y = \frac{a\sqrt{b} + b\sqrt{a}}{2} \quad \text{где} \quad a = \begin{cases} e^x, & \text{если } b > 0 \\ 0, & \text{если } b = 0 \\ \sin x, & \text{в остальных случаях} \end{cases}$$

3. Построить схему алгоритма

$$y = ax^2 + b, \text{ где } a = \begin{cases} e^x, & \text{если } x < b \\ \sin x, & \text{если } x = b \\ \log_{\sqrt{b}}(\sin x), & \text{если } x > b \end{cases}$$

4. Построить схему алгоритма

$$y = \begin{cases} 1 - e^{ax} \sin(ax + b), & \text{при } x > \pi \\ 1 - e^{-ax}(ax + b), & \text{при } -\pi \leq x \leq \pi \\ 1 - (e^{-ax} + e^{-bx}), & \text{при } x < -\pi \end{cases}$$

$$\text{где } x = ab^2 - \sin b$$



5. Построить схему алгоритма

$$y = \begin{cases} \frac{1}{\sqrt{d}} \operatorname{arctg} \frac{b+c}{\sqrt{d}}, \text{ при } d > 0 \\ -\frac{1}{b+c}, \text{ при } d = 0 \\ \frac{1}{2\sqrt{d}} \ln \sqrt{d}, \text{ при } d > 5 \end{cases} \quad \text{где } d = ac - b^2$$

6. Построить схему алгоритма

$$n = \begin{cases} 1, \text{ если } x \geq 0 \text{ и } y \geq 0 \\ 2, \text{ если } x < 0 \text{ и } y \geq 0 \\ 0, \text{ в остальных случаях} \end{cases} \quad \text{где } y = \sqrt[3]{\operatorname{tg} x^2 + \sqrt{\sin x + \cos x}}$$

7. Построить схему алгоритма

$$y = \frac{1}{x} + \frac{\sqrt{e^x + e^z}}{x} \quad \text{где } z = \begin{cases} x^3, \text{ если } x > b \\ 0, \text{ если } x < b \\ 7x, \text{ если } x = b \end{cases}$$

8. Построить схему алгоритма: $\sum \frac{1}{(2i)^2}$

9. Дано натуральное n . Построить схему алгоритма: $\sum_{k=1}^n \frac{1}{K}$

10. Дано натуральное n . Построить схему алгоритма: $\sum_{k=1}^n \frac{1}{K^5}$

11. Дано натуральное n . Построить схему алгоритма:

$$\sum_{k=1}^n \frac{1}{(2k+1)^2}$$

12. Задан одномерный массив из n элементов. Найти среднее арифметическое всех элементов массива.



Информатика

13. Задан одномерный массив из n элементов. Найти наименьший элемент в массиве.
14. Задан одномерный массив из n элементов. Найти количество положительных элементов массива.
15. Задан одномерный массив из n элементов. Найти количество отрицательных элементов массива.
16. Задан одномерный массив из n элементов. Определить, сколько раз встречается число 7 среди элементов массива.
17. Задан одномерный массив из n элементов. Определить, сколько элементов массива меньше, чем число 6.
18. Задан одномерный массив из n элементов. Определить, сколько элементов массива больше, чем число 3.
19. Задан одномерный массив из n элементов. Найти сумму всех неотрицательных элементов массива.
20. Задан двумерный массив. Найти сумму всех элементов массива, имеющих четные индексы.
21. Задан двумерный массив. Найти наибольший из элементов массива, имеющих нечетные индексы.
22. Задан двумерный массив. Найти среднее арифметическое всех положительных элементов массива.
23. Задан двумерный массив. Найти среднее арифметическое всех отрицательных элементов массива.
24. Задан двумерный массив. Найти сумму элементов массива, превышающих число 5.
25. Задан двумерный массив. Найти наибольший элемент в третьем столбце матрицы.
26. Задан двумерный массив. Расположить все элементы матрицы в строку в порядке возрастания.
27. Задан двумерный массив. Найти сумму всех положительных элементов матрицы.



СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. А.В. Басова, О.В. Смирнова и др. Краткий курс информатики, Ростов н/Д. ЮФУ, 2008г.
2. Ю.А.Стоцкий. Самоучитель. Office XP. — Питер, 2003г.
3. С.В. Симонович. Информатика. Базовый курс. — Питер, 2002г.
4. О.Э. Згадзай, С.Я. Казанцев, Л.А. Казанцева. Информатика для юристов. — Москва, 2001г.
5. Н. Угринович. Информатика и информационные технологии — М., БИНОМ, 2003г.
6. Немнюгин С.А. Turbo Pascal. СПб: Питер, 2000.
7. Информатика. Базовый курс/ Под ред. С.В. Симонович – СПб: Издательство «Питер», 2000. – 640 с.
8. Информатика: Учебник. /Под ред. Н.В. Макаровой. – М: Финансы и статистика, 2001. – 768 с.
9. Фигурнов В.Э. IBM PC для пользователя. — М.: Информ. — М., 2000 — 432 с.
10. Р. Хершель. TURBO PASCAL — М.: МП «МИК», 1991. — 342 с.



ЛАБОРАТОРНАЯ РАБОТА №2 «ПРОГРАММИРОВАНИЕ АЛГОРИТМОВ РАЗВЕТВЛЯЮЩИХСЯ И ЦИКЛИЧЕСКИХ СТРУКТУР»

Содержит общие сведения программировании на языке Pascal ABC, позволяет освоить: основные приемы построения программ для решения различных задач.

1. ЦЕЛЬ РАБОТЫ

Целью работы является ознакомление студентов с организацией программ разветвляющихся и циклических типов, а также закрепление практических навыков работы в программировании.

2. ПОРЯДОК ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ

Для успешного выполнения данной лабораторной работы студент должен:

- изучить теоретическую часть (раздел 3);
- согласно индивидуальному варианту составить схему алгоритма решения задачи;
- в соответствии со схемой составить программу решения задачи;
- выполнить решение задачи на персональной ЭВМ.

По окончании работы на ЭВМ необходимо должным образом оформить отчет, представить его преподавателю и защитить свою работу.

3. КРАТКИЕ СВЕДЕНИЯ ИЗ ТЕОРИИ

В большинстве встречающихся задач требуется использование определенного набора типовых приемов алгоритмизации, к которым можно отнести алгоритмы разветвляющейся и циклической структур.

В алгоритмах разветвляющейся структуры предусмотрено разветвление последовательности действия на несколько направлений в зависимости от итога проверки заданного условия.



В алгоритме циклической структуры предусмотрено неоднократное выполнение одной и той же последовательности действий.

3.1. Программирование алгоритмов разветвляющейся структуры

В языке ПАСКАЛЬ для программирования разветвляющейся структуры используются операторы безусловного перехода, условного перехода и оператор выбора.

Оператор безусловного перехода имеет вид `GOTO m`; где *m* - метка, представляющая собой целое число без знака. Метка должна быть описана в разделе описания. Этот оператор передает управления к оператору, описанному меткой *m*. Причем *m* может быть помечен пустой оператор, который не предписывает никаких действий и используется для выхода.

Условный оператор имеет следующие виды записи:

А) `IF B THEN A`;

Б) `IF B THEN A ELSE C`; где *B* - выражение логического типа; *A*, *C* - простые или сложные операторы.

При использовании условного оператора вида а) будет организовано вычисление оператора *A*, если логическое выражение *B* принимает значение TRUE (истина). Если же оно принимает значение FALSE (ложь), то будет выполняться оператор, который следует за этим условным оператором.

Пример 1. Пусть даны два числа *X* и *Y*. Если первое больше второго по абсолютной величине, то необходимо уменьшить первое в 5 раз. Иначе оставить числа без изменения.

Блок-схема алгоритма и программа имеют вид:

```
PROGRAM A;
VAR X, Y: REAL ;
  BEGIN
  WRITELN (' ВВЕСТИ X,Y');
  READ (X,Y);
  IF ABS(X)> ABS(Y) THEN X:= X/5;
  RITELN (' X=',X,' ', 'Y='Y);
  END.
```

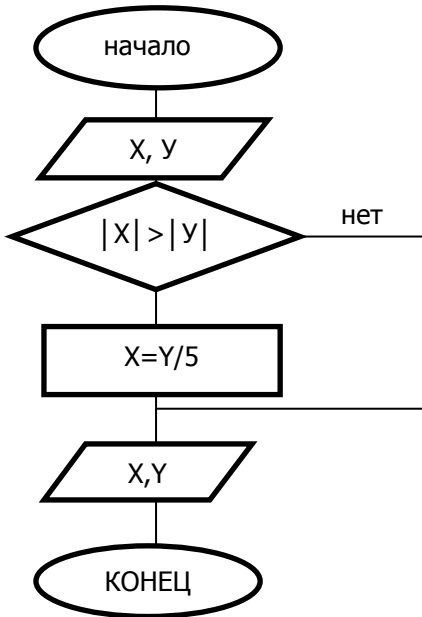


Рис.1

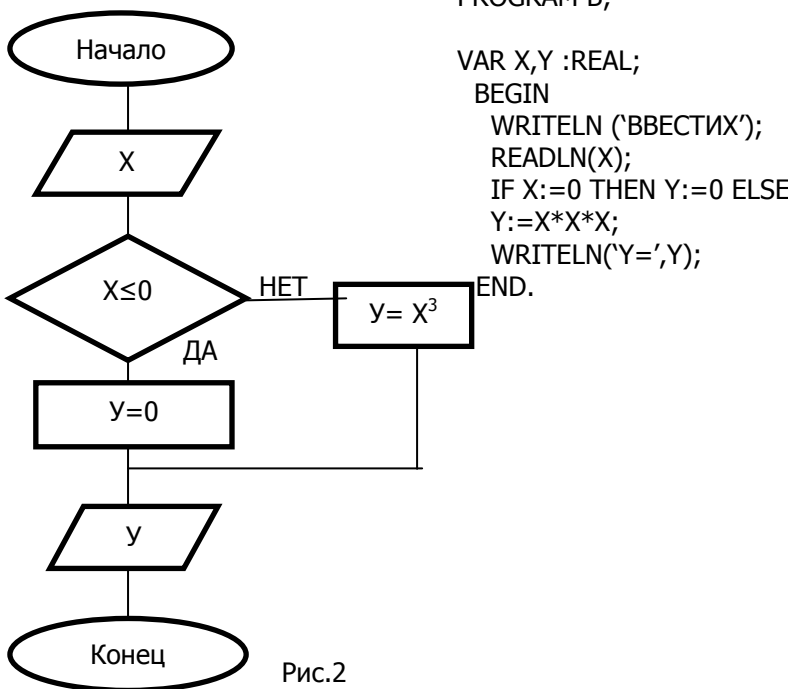
При использовании условного оператора вида б) будет организовано вычисление оператора А, если логическое выражение В принимает значение TRUE. Если логическое выражение принимает значение FALSE, то выполняется оператор С, стоящий после ELSE.

Пример 2. Вычислить Y по значениям X, если

$$y = \begin{cases} 0, & \text{если } X \leq 0; \\ X^3, & \text{если } X > 0; \end{cases}$$



Схема алгоритма и программа имеют вид:



В случае, если необходимо выполнение операторов при тех или иных условиях, то это осуществляется объединением операторов в блок, формируемый операторами BEGIN и END

Например, IF B THEN BEGIN A1;A2;A3 END
ELSE BEGIN A4;A5; END;

В этом примере имена с A1 по A5 символически обозначают операторы. Если логическое выражение B принимает значение TRUE, то выполняются операторы A1,A2,A3, в противном случае выполняются операторы A4 и A5. Операторы, заключенные между BEGIN и END, образуют так называемый составной оператор, а сами операторы BEGIN и END часто именуют операторными скобками.

**Пример 3.**

Требуется установить значения переменных C и D в зависимости от состояния переменных A и B , причем, если значение переменной A равно B , то переменной C присваивается значение, равное сумме значений переменных A и B , а переменной D - значение переменной B . В случае, если значение переменной A не равно значению переменной B , то переменной C присваивается значение, равное разности значений переменных A и B , а переменной D - значение переменной B , возведенное в квадрат.

Схема алгоритма и программа имеют следующий вид:

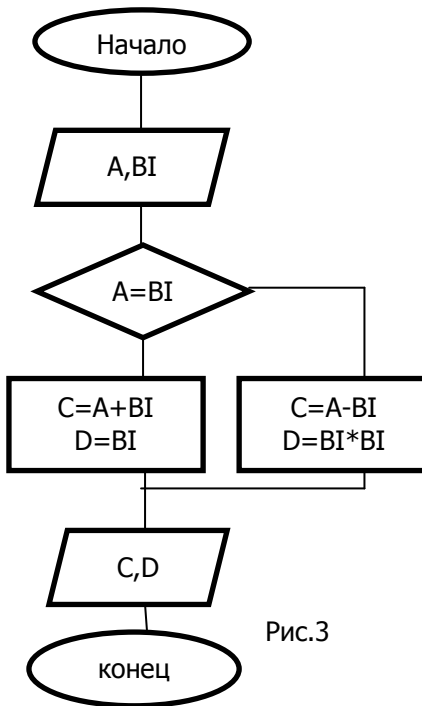


Рис.3

```

PROGRAMC(INPUT, OUTPUT);
  LABEL IØ;
  VAR A,B,I,C,D:REAL;
BEGIN
  WRITELN('ВВЕДИ A,B');
  READ(A, B);
  IF A=B THEN BEGIN
    C=A+B;
    D=B;
    GOTO IØ;
  ELSE BEGIN
    C=A-B;
    D=B*B;
    END;
  IØ: WRITE('C=',C,' ',D=',D');
  END.

```

Оператор выбора варианта имеет вид:

```
CASE C OF NI : AI;N2:A2;...NN:ANEND;
```

Где C -выражение скалярного типа (но не вещественного);
 $N1, N2, \dots, NN$ –метка операторов; $A1, A2, \dots, AN$ – операторы.

Выполнение оператора начинается с вычисления значения выражения C . Затем выполняется тот оператор, метка которого совпала с полученным значением. Если же вычисленное значение выражение C не совпало ни с одной из меток, то предусмотр-



рено появление сообщения об ошибке.

Пример 4. Пусть требуется напечатать значение введенной с клавиатуры цифры, принимающей значение от единицы до пяти, с контролем введенного значения.

Программа имеет вид:

```
PROGRAM VES;  
  VAR I: INTEGER;  
BEGIN  
  WRITELN( 'ВВЕСТИ I');  
  READ(I);  
  WRITELN( 'I=',I);  
  CASE I OF  
    1: WRITELN( ' ВЫ ВВЕЛИ => 1');  
    2: WRITELN( ' ВЫ ВВЕЛИ =>2');  
    3: WRITELN( ' ВЫВВЕЛИ =>3');  
    4,5: WRITELN( 'ВЫ ВВЕЛИ=>4 ИЛИ 5');  
  END;  
END.
```

3.2. Программирование алгоритмов циклической структуры

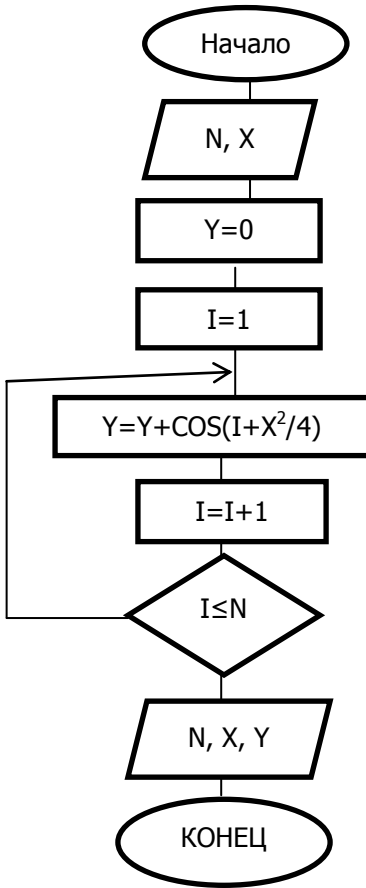
Программы циклической структуры могут быть организованы с помощью следующих операторов IF, FOR, WHILE, REPEAT. Условный оператор IF требует задания начального параметра цикла, проведения в цикле вычисления с этим параметром, изменения его на величину шага и проверки условия окончания цикла. Если цикл не закончен, то вычисления продолжаются.

Пример 5.

Составить схему алгоритма и программу для вычисления выражения.



Схема алгоритма и программа имеют вид:



```

PROGRAM SUMIP;
  LABEL 3;
  VAR  X,Y: REAL;
  I,N:INTEGER;
  BEGIN
    WRITELN( 'ВВЕСТИ N,X');
    READ(N, X);
    Y:=0;
    I:=1;
    3: Y:=Y+COS(I+SQR(X)/4);
    I:=I+I;
    IF X<=N THEN GOTO 3;
    WRITELN( 'N=',N);
    WRITELN('X=',X, '      ',Y=',Y);
    END.
  
```

Рис.4

Оператор цикла FOR используется для организации цикла с известным числом повторений. При этом различают две формы записи:

А) с организацией счета при изменении параметра цикла от начального (меньшего) значения до конечного (большого) значения этого параметра цикла. Такой оператор цикла имеет вид:

```
FOR K:=NZ TO KZ DO
```

Б) с организацией счета при изменении параметра цикла от начального(большого) значения до конечного(меньшего) значения этого параметра. Этот оператор имеет вид:



```
FOR K:= NZ DOWNTO KZ DO
```

Где K- параметр цикла (целочисленная переменная), NZ, KZ- выражения, задающие соответственно начальное и конечное значения параметра цикла;
S—простой или составной оператор (тело цикла).

Оператор цикла выполняется следующим образом. Параметру цикла K присваивается начальное значение NZ. Затем уравнение передается в тело цикла и выполняется оператор S, после выполнения которого параметр цикла меняет свое значение на единицу(шаг изменения параметра цикла). При этом шаг равен +I, если используется оператор цикла с ключевым словом TO(случай а), а если используется оператор цикла с ключевым словом DOWNTO (случай б), то шаг равен -I. Далее измененное значение параметра цикла сравнивается с конечным значением KZ и, если параметр цикла не превышает K(случай а) или превышает KZ (случай б), то управление передается в тело цикла и выполняется оператор S; в противном случае осуществляется выход из цикла.

Рассмотрим фрагменты записи операторов цикла:

```
VARY,I: INTEGER;
```

```
1) Y:=0; FOR I=3 TO 5 DO Y=Y+1;
```

```
2) Y:=0; FOR I=12 DOWNTO 6 DO Y:=Y+1;
```

В результате выполнения первого оператора цикла параметр цикла I будет изменяться от начального значения I=3 до конечного значения I=5 с шагом, равным +I, причем после окончания цикла значения переменной Y=3. При выполнении второго оператора цикла параметр цикла I будет изменяться от начального значения I=12 до конечного значения I=6 с шагом, равным -I. После окончания цикла Y=7.

Пример 6.

Используя условия примера 5 программу с изменением оператора цикла FOR.

```
PROGRAM SUM;
VAR X,Y: REAL;
I,N: INTEGER;
BEGIN
  WRITELN( 'ВВЕСТИ N,X ');
  READ(N, X);
  Y:=0;
  FOR I:=1 TO N DO
    Y:=Y+COS (I+SQR(X)/4);
```



```
WRITELN ('N=', N , ' ', 'X=',X);
WRITELN('Y=',Y);
```

```
END.
```

Оператор цикла WHILE позволяет организовать цикл, количество повторений которого зависит от включенного в него условия, т.е. цикл с неизвестным числом повторений. Этот оператор имеет вид WHILE LV DO

Где LV- логическое выражение;

S- простой или составной оператор.

Выполнение оператора начинается с проверки выражения LV. Если логическое выражение имеет значение TRUE, то выполняется оператор S. Выполнение оператора S продолжается до тех пор, пока значение выражения LV не станет равным FALSE. В этом случае уравнение передается оператору, следующему за оператором S. Если же выражение LV имеет значение FALSE при первоначальной проверке, то оператор S не выполняется ни разу. При этом, чтобы выйти из цикла, внутри его должно быть организовано изменение значений переменных, входящих в логическое выражение. Иначе оператор цикла будет выполняться бесконечное число раз.

Пример 7.

Составим программу, используя условия примера 5. Программа запишется в виде:

```
PROGRAM SUMWHL;
VARX,Y : REAL;
I,N : INTEGER;
BEGIN
WRITELN('ВВЕСТИ N,X');
READ(N,X);
Y:=0;
I:=1;
WHILE I<=N DO
BEGIN
Y=Y+ COS( I+SQR(X)/4);
I:=I+1;
END;
WRITELN( 'N=',N,' ', 'X=',X);
WRITELN('Y=',Y);
END.
```



Оператор цикла REPEAT также позволяет организовать цикл с неизвестным числом повторений. Такой оператор имеет вид:

```
REPEATS  
UNTILLV
```

где S – простой или составной оператор;
LV- логическое выражение.

Выполнение оператора REPEAT начинается с вычисления оператора S и продолжается до тех пор, пока не выполняется LV т.е. когда значение этого логического выражения равно FALSE. Это означает, что проверка LV проводится после каждой итерации и в случае принятия LV значения TRUE осуществляется выход из цикла. При использовании составного оператора S операторные скобки (BEGIN и END) не требуются. Кроме того оператор, стоящий перед словом UNTIL, не имеет после себя точку с запятой.

Пример 8.

Задан массив вещественных чисел, состоящий из десяти элементов. Требуется вывести четные элементы этого массива. Программа имеет вид:

```
PROGRAM MAS;  
VAR I: INTEGER;  
  
A: ARRAY[1...10] OF REAL;  
BEGIN  
  WRITELN(' ВВЕСТИМАССИВ(1..10)');  
  FOR I:=1 TO 10 DO;  
    READ (A[I]);  
    I:=0;  
    REPEAT  
      I=I+2;  
      WRITE (A[I]:5:2)  
    UNTIL I>10;  
END.
```



3.3 Программирование алгоритмов со структурой вложенных циклов.

Для реализации сложных алгоритмов возникает необходимость организовать в составе одного цикла еще несколько других циклов. При этом цикл, внутри которого находятся другие циклы, называется внешним, а вложенные в него - внутренними. Алгоритмы циклических вычислительных процессов, содержащие внутренние циклы, будем называть алгоритмами со структурой вложенных циклов.

Организовывать вложенные циклы необходимо таким образом, чтобы циклы не пересекались, т.е., чтобы тело внутреннего цикла полностью размещалось в теле внешнего цикла.

Изменение параметров внутреннего и внешнего циклов происходит не одновременно, следующим образом:

- при начальном значении параметра внешнего цикла параметра внутреннего цикла последовательно принимает все возможные значения(от своего начального значения до конечного), причем передача управления из внутреннего во внешний цикл происходит после принятия параметром внутреннего цикла своего конечного значения и выполнения тела этого цикла;
- происходит изменение параметра внешнего цикла на величину шага, а параметр внутреннего цикла опять последовательно принимает все свои значения;
- выход из внешнего цикла производят тогда, когда параметр этого цикла примет свое конечное значение, а параметр внутреннего цикла последовательно примет все свои значения.

Пример 8.

Составить программу для вычисления

$$P = \sum_{n=1}^m \sqrt{2 * S / n} * \prod_{i=1}^n N / \sqrt{i}$$

```
PROGRAM VLZ;
  VAR  N,M, I : INTEGER;
       S,R,P,Y,Z: REAL;
BEGIN
  WRITELN ('ВВЕСТИ M,S');
  READ (M,S);
  R:=0;
```



```
FOR N:=1 TO M DO
  BEGIN
    P:=1;
    FOR I:=1 TO N DO
      BEGIN
        Y:=N/SQRT(I);
        P:=P*Y;
      END;
      Z:=SQRT(2*S/N)*P;
      R:=R+Z;
    END;
  WRITELN( 'R=',R);
END.
```




4. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Опишите оператор безусловного перехода.
2. Опишите конструкцию условного оператора.
3. Для какой цели используются операторные скобки?
4. Опишите оператор выбора варианта.
5. С использованием каких операторов могут быть организованы циклические структуры?
6. В каких случаях используется оператор цикла со служебным словом DOWNTO?
7. Каких этапов требует организация цикла с помощью условного оператора?
8. С помощью каких операторов можно организовать цикл с неизвестным числом повторений?
9. Что означают служебные слова TRUEи FALSE?
10. Как изменяются параметры внешнего и внутреннего циклов?



5. ИНДИВИДУАЛЬНЫЕ ЗАДАНИЯ

1. Вычислить значение

$$Z = \begin{cases} X^2 + 3, & \text{если } x < 0, \\ X - 0,8, & \text{если } x > \pi/2, \\ \cos X, & \text{если } 0 \leq X \leq \pi/2. \end{cases}$$

2. Вычислить значение

$$Z = \begin{cases} a - b, & \text{если } a > 0, \\ a + b, & \text{если } a = 0, \\ ab, & \text{если } a < 0 \end{cases}$$

3. Вычислить значение

$$Z = \begin{cases} a/b, & \text{если } a > 0, \\ a * b, & \text{если } a = 0, \\ a - b, & \text{если } a < 0 \end{cases}$$

4. $Z = a + b \sin x / x^2 - bx + 0,5$, если $x \geq 0$

5. Вычислить значение

$$Z = \begin{cases} X^2 + Y^2, & \text{если } X^2 + Y^2 < 1, \\ X + Y, & \text{если } X^2 + Y^2 > 1, \\ 0,8, & \text{если } X^2 + Y^2 = 1. \end{cases}$$

6. Вычислить значение

$$Y = \begin{cases} a + bx + cx^2, & \text{если } K = 1, \\ d + cx + fx^2, & \text{если } K = 2, \\ g + hx + tx^2, & \text{в остальных случаях} \end{cases}$$



7. Вычислить значение

$$Y = 18,9X^2 - 1,6X + 1,97, \text{ если } X < 0 \text{ и } Z = 9,75X^2 - 0,35 \operatorname{arctg} x, \text{ если } X \geq 0.$$

8. Вычислить значение

$$Z = a + b \sin x / x^2 - bx + 0,5, \text{ если } x \geq 0.$$

9. Вычислить значение

$$Z = \sin x / x^2 - \sqrt{1+3x}, \text{ если } x \leq 0,5$$

10. Вычислить значение

$$Z = \begin{cases} \operatorname{Arctg} x + y/1 - xy, & \text{если } xy < 1, \\ 3,14, & \text{если } xy = 1, \\ 3,14 + (x + y/1 - xy), & \text{если } xy > 1. \end{cases}$$

11. Вычислить значение

$$Z = \begin{cases} Y^2 - 0,3, & \text{если } y < 0, \\ 1, & \text{если } 0 \leq y \leq 1, \\ Y^2 + 0,3, & \text{если } y > 1. \end{cases}$$

12. Вычислить значение

$$A_i = X^i / (2i), \quad i = 1, 2, \dots, 30.$$

13. Вычислить значение

$$A_i = (X^{2i} \operatorname{SIN}(X^i)) / i^2, \quad i = 1, 2, \dots, 30.$$

14. Вычислить значение

$$A_i = ((-1)^i * X^i) / (i(i+1)(i+2)), \quad i = 1, 2, \dots, 30.$$

15. Вычислить значение

$$A_i = (X^{2i} * (i+3)) / 2i, \quad i = 1, 2, \dots, 30.$$

16. Вычислить значение



$$A_i = (X^i * (i+2)) / i + 1, \quad i=1, 2, \dots, 30.$$

17. Даны вещественные числа X, B и целое K ($1 \leq K \leq 4$). Вычислить $z = 5Y^2 - b$, где Y вычисляется по формулам:

$$1) Y = 2X \quad 2) Y = X^2 - 3,5 \quad 3) Y = X + 9,6$$

4) $Y = X^2 / 6$, а значение X трактуется как номер для вычисления Y .

18. Даны вещественные числа Y, C и целое n ($1 \leq n \leq 5$). Вычислить $Z = 3 \cos(X) / X + C$, где X вычисляется по формулам:

$$1) X = 1,2 Y^2 \quad 2) X = 3,5 Y \quad 3) X = 1,5$$

4) $X = \arctg(Y)$ 5) $X = 6 + 0,5 Y^2$, а значение n трактуется как номер для вычисления X .

19. Дано целочисленное значение n и вещественные значения X, Y и Z . При $n < 0, n = 0$ или $n > 0$ изменить соответственно значения X на $X - 0,5$, Y на $Y / 3,5$, а Z на Z^2 .

20. Вычислить значение $Y_i = X_i + \sin x_i$ для всех $0 \leq X_i \leq 1$, если $X_i = X_{i-1} + h$, $X_0 = 0$ и $h = 0,1$

21. Все отрицательные компоненты вещественного вектора $a(a_1, a_2, \dots, a_{10})$ заменить их квадратами.

22. Дан вещественный вектор $X(X_1, X_2, \dots, X_{10})$. Умножить на 2 компонента с нечетными номерами.

23. Дан вещественный вектор $b(b_1, b_2, \dots, b_{10})$. Все компоненты с четными номерами уменьшить на 0,5.

24. Все компоненты вещественного вектора $C(C_1, C_2, \dots, C_{10})$, начиная с первой по порядку положительной компоненты увеличить в два раза.

25. Вычислить $n!$ для $n=7$.

26. Присвоить переменной T значение "3", если числа X, Y, Z равны "2", если числа равны; "1", если все числа различны.



27. Определить количество тех членов последовательности

$A_k = \sin(K^2 + C)$, $K=1, 2, \dots, 10$, которые удовлетворяют условию $0 \leq A_k \leq 1/2$.

28. Определить значения последовательности $\{x_i\}$ из условия $x_{i+1} = 0,5x_{i-1} - x_i^2$, $x_1 = x_i = 0,65$ для $i=3, 4, \dots, 10$.

29. Вычислить значение

$$Y = X^n / 3! \text{ Для } n=1, 2, \dots, 10$$

30. Вычислить значение

$$Y = (1)^n / 2n(2n-1) \text{ для } n=1, 2, \dots, 10.$$

31. Вычислить значение

$$Y = 2n/n^2 \sqrt{x+n} \text{ для } n=1, 2, \dots, 10.$$



СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. А.В. Басова, О.В. Смирнова и др. Краткий курс информатики, Ростов н/Д. ЮФУ, 2008г.
2. С.В. Симонович. Информатика. Базовый курс. — Питер, 2002г.
3. Немнюгин С.А. Turbo Pascal. СПб: Питер, 2000.
4. Информатика. Базовый курс/ Под ред. С.В. Симонович – СПб: Издательство «Питер», 2000. – 640 с.
5. Р. Хершель. TURBO PASCAL — М.: МП «МИК», 1991. — 342 с.



ЛАБОРАТОРНАЯ РАБОТА №3 «СИСТЕМЫ СЧИСЛЕНИЯ»

Содержит общие сведения о системах счисления, позволяет освоить: способы перевода из одной системы счисления в другую, основные приемы арифметических действий в различных системах счисления.

1. ЦЕЛИ РАБОТЫ

В результате прохождения занятия студент должен:

- **знать:**
 1. основные определения (система счисления, основание системы счисления, база системы);
 2. примеры позиционных и непозиционных систем счисления;
 3. правила перевода из одной системы счисления в другую;
 4. правила выполнения арифметических операций с системами счисления
- **уметь:**
 1. применять правила перевода из одной системы счисления в другую;
 2. выполнять арифметические операции над системами счисления.

Для успешного выполнения лабораторной работы по данной теме студент должен:

- изучить теоретическую часть;
- выполнить предложенные варианты заданий

По окончании работы студент должен представить отчет по выполнению лабораторной работы.

В отчете о проделанной работе должны быть раскрыты следующие вопросы:

- тема и цель занятия;
- индивидуальное задание;
- порядок выполнения индивидуального задания (фактическое выполнение);
- полученные результаты.

Время выполнения работы – 4 часа



2. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Под **системой счисления** понимается способ представления любого числа с помощью некоторого алфавита символов, называемых цифрами.

Различают два типа систем счисления:

- **непозиционные**, когда значение цифры в числе не зависит от ее места в записи числа;
- **позиционные**, когда значение каждой цифры числа определяется ее позицией в записи числа.

Примером непозиционной системы счисления является римская система. В этой системе цифры обозначаются буквами латинского алфавита:

$$\begin{array}{llll} I = 1 & V = 5 & X = 10 & L = 50 \\ C = 100 & D = 500 & M = 1000 & \end{array}$$

Пример 1. Записать число 444 в римской системе.

Решение: $444 = 400 + 40 + 4 = CD + XL + IV = CDXLIV$

К недостаткам таких систем относятся наличие большого количества знаков и сложность выполнения арифметических операций.

Наибольшее распространение получили позиционные системы счисления, в которых значение любой цифры определяется не только конфигурацией ее символов, но и местоположением – позицией, которое она занимает в числе. При этом под основанием позиционной системы счисления q понимается значение равное числу знаков, используемых для отображения цифр в данной системе счисления. В общем виде число A с основанием q в позиционной системе счисления можно представить выражением:

$$A_{(q)} = a_m q^m + a_{m-1} q^{m-1} + \dots + a_1 q^1 + a_0 q^0 + a_{-1} q^{-1} + a_{-2} q^{-2} + \dots + a_{-n} q^{-n} \quad (1)$$

где a_i – цифры системы счисления, m, n – число целых и дробных разрядов соответственно.



Десятичная система счисления

В десятичной системе счисления $q=10$ и набор символов включает цифры $0,1,2,3,4,5,6,7,8,9$. Таким образом, по формуле (1) число $478,53$ в десятичной системе счисления можно представить в виде:

$$478,53_{(10)} = 4 \cdot 10^2 + 7 \cdot 10^1 + 8 \cdot 10^0 + 5 \cdot 10^{-1} + 3 \cdot 10^{-2}$$

По такому принципу можно строить различные системы счисления.

Двоичная система счисления

Двоичная система счисления – это позиционная система счисления с основанием $q=2$ и символами отображения чисел, состоящими из двух цифр: 0 и 1 .

В соответствии с формулой (1) запись числа $10111,101_{(2)}$ соответствует следующему числу в десятичной системе счисления:

$$10111,101_{(2)} = 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} = 23,625_{(10)}$$

Восьмеричная система счисления

Это позиционная система счисления с основанием $q=8$ и символами отображения чисел, состоящими из восьми цифр: $0,1,2,3,4,5,6,7$. Число восемь десятичной системы счисления записывается в виде $10(8)$, так как является единицей следующего разряда. Числу $478,53(10)$ будет соответствовать число $736,41(8)$, так как

$$736,41(8) = 7 \cdot 82 + 3 \cdot 81 + 6 \cdot 80 + 4 \cdot 8^{-1} + 1 \cdot 8^{-2} = 478,53(10)$$

Шестнадцатеричная система счисления

Это позиционная система счисления с основанием $q=16$ и символами отображения чисел, состоящими из десяти цифр: $0,1,2,3,4,5,6,7,8,9$ и шести букв: A – соответствует числу $10_{(10)}$, B – числу $11_{(10)}$, C – числу $12_{(10)}$, D – числу $13_{(10)}$, E – числу $14_{(10)}$, F – числу $15_{(10)}$. Число 16 в десятичной системе счисления записывается в виде $10_{(16)}$, а число, например $1DE,87_{(16)} = 1 \cdot 16^2 + 13 \cdot 16^1 + 14 \cdot 16^0 + 8 \cdot 16^{-1} + 7 \cdot 16^{-2} = 478,53(10)$



Правило перевода из десятичной системы счисления в недесятичную

Перевод чисел из одной системы счисления в другую составляет важную часть машинной арифметики. Рассмотрим основные правила перевода из десятичной системы счисления в недесятичную.

Правило перевода целой части числа

Целую часть десятичного числа необходимо последовательно делить на основание q системы счисления, в которой переводят, до тех пор пока частное не будет меньше делителя. Полученный результат записывается как последовательность последнего частного и остатков от деления в обратном порядке.

Правило перевода дробной части числа

Дробную часть десятичного числа необходимо последовательно умножать исходную дробь и дробные части, получаемых произведений, на основание q системы счисления, в которой переводят. Процесс перевода заканчивается, если дробная часть получилась равной нулю или если вычислено заданное количество цифр после запятой. Полученная дробь записывается в виде последовательности целых частей произведений, начиная с первого.

Пример 2. Перевести $165_{(10)}$ в двоичную систему счисления.

Решение:

$$165 \underset{\cdot}{\div} 2$$

$$\underline{16} \quad 82 \underset{\cdot}{\div} 2$$

$$5 \quad \underline{8} \quad 41 \underset{\cdot}{\div} 2$$

$$\underline{4} \quad 2 \quad \underline{40} \quad 20 \underset{\cdot}{\div} 2$$

$$\underline{1} \quad \underline{2} \quad \underline{1} \quad \underline{20} \quad 10 \underset{\cdot}{\div} 2$$

$$\underline{0} \quad \underline{0} \quad \underline{10} \quad 5 \underset{\cdot}{\div} 2$$

$$\underline{0} \quad \underline{4} \quad 2 \underset{\cdot}{\div} 2$$

$$\underline{1} \quad \underline{2} \quad \underline{1}$$

$$\underline{1}$$



$$\begin{array}{r}
 0,25 \\
 * \\
 \hline
 0,50 \\
 * \\
 \hline
 1,00
 \end{array}$$

Ответ: $165,25_{(10)} = 11100101,01_{(2)}$

Правила арифметических операций в различных системах счисления

Арифметические операции над недесятичными системами счисления выполняются по тем же правилам, что и в десятичной системе счисления.

Приведем несколько примеров сложения и вычитания двоичных и шестнадцатеричных чисел:

Пример 3.

$$a) 78_{(10)} = 4E_{(16)} = 100 \ 1110_{(2)}$$

$$+ \quad \quad + \quad \quad +$$

$$47_{(10)} = 2F_{(16)} = 10 \ 1111_{(2)}$$

$$\hline \hline \hline$$

$$125_{(10)} = 7D_{(16)} = 111 \ 1101_{(2)}$$

$$b) 1450_{(10)} = 5AA_{(16)} = 101 \ 1010 \ 1010_{(2)}$$

$$- \quad \quad - \quad \quad -$$

$$427_{(10)} = 1AB_{(16)} = 1 \ 1010 \ 1011_{(2)}$$

$$\hline \hline \hline$$

$$1023_{(10)} = 3FF_{(16)} = 11 \ 1111 \ 1111_{(2)}$$

В десятичной системе при изображении чисел, больших девяти, различные символы (т. е. цифры 0, 1, ..., 9) располагаются друг за другом, например 365. Комбинации этих символов могут быть получены сложением 1 и 0 с последующим добавлением 1 к каждой получаемой сумме: $1 + 0 = 1$, $1 + 1 = 2$, $1 + 2 = 3$ и т. д. Поскольку при операции сложения $1 + 9$ сумму невозможно изобразить одним символом, поэтому слева от цифры девять, к которой прибавляется 1, ставится 1, а саму цифру девять заменяют цифрой 0, иначе говоря, осуществляют перенос в старший



разряд. После этого можно продолжать операцию добавления 1 к сумме. Описанный процесс, называемый счетом, позволяет получить все комбинации цифр, используемых для изображения чисел в десятичной системе.

- Заметим, что при счете в десятичной системе особое внимание следует обращать на выполнение переноса и замену наибольшей цифры наименьшей.

Рассмотрим теперь, как происходит счет в двоичной системе счисления. Напомним, что в этой системе для изображения чисел используются только два символа: 0 и 1. Начнем счет также, как и в десятичной системе, складывая 1 и 0. Естественно, что $1 + 0 = 1$. Добавим к полученной сумме 1. Поскольку сумму в двоичной системе невозможно представить одной цифрой, как и раньше, выполним перенос: припишем слева к первой сумме 1, а ее значение заменим на 0.

- Заметим, что операция переноса выполняется также, как при счете в десятичной системе с той лишь разницей, что в двоичной системе используются только два символа (две двоичные цифры).

В шестнадцатеричной системе счисления используются 16 символов: цифры от 0 до 9 и буквы от A до F. Поэтому шестнадцатеричное число может иметь вид 03F4. Чтобы определить шестнадцатеричные числа, можно вновь повторить процесс счета подобному тому, как это делалось в случае десятичной и двоичной систем. Сложение и умножение чисел в двоичной, восьмеричной и шестнадцатеричной систем счисления иллюстрируют таблицы 1-6, где по вертикале в первом столбце и по горизонтали в первой строке записаны цифры, на пересечении столбцов и строк результат сложения или умножения.

Таблица 1

Сложение чисел в двоичной системе счисления

+	0	1
0	0	1
1	1	10



Таблица 2

Умножение чисел в двоичной системе счисления

×	0	1
0	0	0
1	0	1

Пример 4.

$$\begin{array}{r} \text{a) } 1010111,1011 \\ + 11001111,10101 \\ \hline 100100111,01011 \end{array}$$

$$\begin{array}{r} 100100111,01011 \\ - 11001111,10101 \\ \hline 1010111,1011 \end{array}$$

$$\begin{array}{r} \text{b) } 1001,11 \\ * 1011,1 \\ \hline 100111 \\ + 100111 \\ 100111 \\ \hline 100111 \\ \hline 1101111,001 \end{array}$$

Таблица 3

Сложение чисел в восьмеричной системе счисления

+	0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7	10
2	2	3	4	5	6	7	10	11
3	3	4	5	6	7	10	11	12
4	4	5	6	7	10	11	12	13
5	5	6	7	10	11	12	13	14
6	6	7	10	11	12	13	14	15
7	7	10	11	12	13	14	15	16



Таблица 4

Умножение чисел в восьмеричной системе счисления

+	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7
2	2	4	6	10	12	14	16
3	3	6	11	14	17	22	25
4	4	10	14	20	24	30	34
5	5	12	17	24	31	36	43
6	6	14	22	30	36	44	52
7	7	16	25	34	43	52	61

Таблица 5

Сложение чисел в шестнадцатеричной системе счисления

+	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10
2	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11
3	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12
4	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13
5	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14
6	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15
7	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16
8	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17
9	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18
A	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19
B	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A
C	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B
D	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C
E	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D
F	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E

**Пример 5.**

$$\begin{array}{r} EF567,AB \\ + 678A4E,567 \\ \hline 767FB6,017 \end{array}$$

$$\begin{array}{r} 767FB6,017 \\ - 678A4E,567 \\ \hline EF567,AB \end{array}$$

Таблица 6

Умножение чисел в шестнадцатеричной системе счисления

+	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2	2	4	6	8	A	C	E	10	12	14	16	18	1A	1E	1C
3	3	6	9	C	F	12	15	18	1B	1E	21	24	27	2A	2D
4	4	8	C	10	14	18	1C	20	24	28	2C	30	34	38	3C
5	5	A	F	14	19	1E	23	28	2D	32	37	3C	41	46	4B
6	6	C	12	18	1E	24	2A	30	36	3C	42	48	4E	54	5A
7	7	E	15	1C	23	2A	31	38	3F	46	4D	54	5B	62	69
8	8	10	18	20	28	30	38	40	48	50	58	60	68	70	78
9	9	12	1B	24	2D	36	3F	48	51	5A	63	6C	75	7E	87
A	A	14	1E	28	32	3C	46	50	5A	64	6E	78	82	8C	96
B	B	16	21	2C	37	42	4D	58	63	6E	79	84	8F	9A	A5
C	C	18	24	30	3C	48	54	60	6C	78	84	90	9C	A8	B4
D	D	1A	27	34	41	4E	5B	68	75	82	8F	9C	A9	B6	C3
E	E	1C	2A	38	46	54	62	70	7E	8C	9A	A8	B6	C4	D2
F	F	1E	2D	3C	4B	5A	69	78	87	96	A5	B4	C3	D2	E1



3. ЗАДАНИЯ ДЛЯ ВЫПОЛНЕНИЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

1. Перевести числа в десятичную систему счисления
2. Перевести следующие числа из десятичной системы счисления в двоичную и восьмеричную системы счисления
3. Перевести следующие числа из десятичной системы счисления в восьмеричную и шестнадцатеричную системы счисления (точность вычислений - 5 знаков после точки)
4. Перевести из восьмеричной системы счисления в двоичную систему счисления
5. Перевести из десятичной системы счисления в шестнадцатеричную
6. Заданы двоичные числа X и Y . Вычислить $X+Y$ и $X-Y$
7. Заданы восьмеричные числа X и Y . Вычислить $X*Y$ и X/Y ,
8. Заданы шестнадцатеричные числа X и Y . Вычислить $X+Y$ и $X-Y$

Номер варианта	Номер задания							
	1	2	3	4	5	6	7	8
1	1101.11 ₍₂₎	84.92	79.47	75.34	89.45	100001.0011 100	75.34 34.75	71.63 32.85
2	1111.01 ₍₂₎	64.68	48.47	25.17	49.38	101101. 11 111.01	25.17 17.25	47.59 71.55
3	100101 ₍₂₎	37.78	57.93	52.23	39.77	101101.011 10.11	52.23 23.52	38.95 44.86
4	1100.01 ₍₂₎	81.39	19.87	63.34	48.29	1001.0011 10.101	63.34 34.63	45.67 75.38
5	43.77 ₍₈₎	67.17	84.67	71.21	53.75	101101.0011 100.0101	71.21 21.71	53.96 31.98
6	37.45 ₍₈₎	41.85	16.87	55.34	32.99	11001.0011 100	55.34 34.55	29.74 81.42
7	AC.25 ₍₁₆₎	23.65	43.54	63.42	74.47	10101.0011 100.1	63.42 42.63	64.38 28.56
8	1101.0011 ₍₂₎	63.47	39.56	44.76	38.79	1001.0011 100.011	76.44 44.76	47.88 51.69



Информатика

9	BC.15 ₍₁₆₎	39.74	23.99	45.65	67.41	10001.0011 100.111	65.45 45.65	67.23 41.69
10	1101.11 ₍₂₎	87.19	46.74	41.73	37.53	1001.0011 100.11	73.41 41.73	39.68 64.97
11	1111.01 ₍₂₎	35.76	43.77	12.73	73.63	1001.0011 10.101	73.12 12.73	57.32 71.23
12	100101 ₍₂₎	88.38	37.85	66.23	74.47	101101.0011 100.0101	66.23 23.66	51.37 38.73
13	1100.01 ₍₂₎	79.47	79.47		68.61	11001.0011 100	65.52 52.65	64.38 28.56
14	43.77 ₍₈₎	48.47	48.47		29.83	10101.0011 100.1	55.54 54.55	47.88 51.69
15	77.45 ₍₈₎	57.93	57.93		56.87	1001.0011 100.011	56.35 34.56	67.23 41.69
16	AB.25 ₍₁₆₎	19.87	19.87		47.97	10001.0011 100.111	55.46 46.55	39.68 64.97
17	1101.0011 ₍₂₎	84.67	84.67		45.63	1001.0011 100.11	66.54 54.66	57.32 71.23
18	BA.15 ₍₁₆₎	16.87	16.87		51.64	100001.0011 100	75.51 51.75	51.37 38.73
19	1100.01 ₍₂₎	32.99	79.47	41.47	51.64	101101. 11 111.01	47.41 41.47	39.68 64.97
20	43.77 ₍₈₎	57.34	48.47	57.23	51.64	101101.011 10.11	57.23 23.57	57.32 71.23



ВОПРОСЫ ДЛЯ САМОКОНТРОЛЯ

1. Почему система счисления называется позиционной?
2. Какие символы содержит система с основанием 8, 16?
3. Чем объяснить широкое применение двоичной системы?
4. Как записывается основание любой системы счисления?
5. Чему всегда равен вес младшего разряда целого числа?
6. Как связан вес старшего разряда целого числа с числом разрядов?
7. Почему первый остаток от деления исходного числа на основание новой системы является младшим разрядом числа в новой системе?
8. На что можно было бы умножать при переводе дроби из некоторой системы счисления в новую систему?
9. Чему равен вес старшего разряда дроби?



ЛИТЕРАТУРА

1. Фигурнов В.Э. IBM PC для пользователя.–М: ИНФРА-М, 2002.– 640с.
2. Целигоров Н.А. Основы информатики. – РГАСХМ, Ростов-н/Д, 2000.
3. Степанов А.Н. Информатика – СПб.: Изд-во «Питер», 2004.