

Глава 8. Алгоритмизация и программирование

Вопросы и задачи

§ 54. Алгоритм и его свойства

Контрольные вопросы

1. Что такое алгоритм?
2. Что такое исполнитель?
3. Чем отличаются формальные и неформальные исполнители?
4. Что такое «система команд исполнителя»? Придумайте исполнителя с некоторой системой команд.
5. Перечислите и объясните свойства алгоритма.
6. Какие существуют способы записи алгоритмов? Какие из них, по вашему мнению, чаще применяются на практике? Почему?

§ 55. Простейшие программы

Контрольные вопросы

1. Сравните структуру программ на языке Паскаль и алгоритмическом языке.
2. Что такое идентификатор?
3. Чем отличаются правила построения имён в алгоритмическом языке и в Паскале?
4. Как записываются комментарии на этих языках? Подумайте, как комментирование можно использовать при поиске ошибок в алгоритме?
5. Сравните операторы вывода на алгоритмическом языке и в Паскале. Как выполняется переход на новую строку?
6. Что такое переменная? Как вы думаете, зачем нужно объявлять переменные?
7. Зачем нужен тип переменной? Почему нельзя записывать в переменную значение другого типа?
8. Какое значение записано в переменной сразу после объявления? Можно ли его использовать?
9. Как задать начальные значения переменных? Сравните алгоритмический язык и Паскаль.
10. Что такое оператор присваивания?
11. Почему желательно выводить на экран подсказку перед вводом данных?
12. Подумайте, когда можно вычислять результат прямо в операторе вывода, а когда нужно заводить отдельную переменную.
13. Что такое форматный вывод? Как вы думаете, где он может быть полезен?

Задачи и задания

1. Используя оператор вывода, постройте на экране следующие рисунки из символов:

```

      Ж      Ж      Ж      Ж      Ж      Ж      Ж
     ЖЖЖ    ЖЖ      Ж  Ж      ЖЖ    ЖЖ  ЖЖ
    ЖЖЖЖЖ  ЖЖЖЖЖЖ  ЖЖЖЖЖ    ЖЖЖ    ЖЖЖЖЖ
     Ж  Ж      ЖЖ    Ж  Ж  Ж      ЖЖЖЖЖ  ЖЖ  ЖЖ
     ЖЖЖ      Ж    ЖЖЖЖЖЖ  ЖЖЖЖЖЖЖЖ  Ж  Ж
  
```

2. Выберите правильные имена переменных (для алгоритмического языка и Паскаля):

| | | | |
|-----|------------|--------|--------------|
| 1 | Vasya | СУ-27 | @mail_ru |
| m11 | Петя | СУ_27 | lenta.ru |
| 1m | Митин брат | _27 | "Pes barbos" |
| m 1 | Quo vadis | СУ(27) | <Ладья> |

3. Пусть **a** и **b** – целые переменные. Что будет выведено в результате работы фрагмента программы:

- | | | |
|----|--|---|
| а) | цел <code>a = 5, b = 3</code> вывод <code>a, 'Z(' , b, ')'</code> | <code>a := 5; b := 3;</code> <code>write(a, 'Z(' , b, ')');</code> |
| б) | цел <code>a = 5, b = 3</code> вывод <code>'Z(a)=' , '(b)'</code> | <code>a := 5; b := 3;</code> <code>write('Z(a)=' , '(b)');</code> |
| в) | цел <code>a = 5, b = 3</code> вывод <code>'Z(' , a, ')=(, a+b, ')'</code> | <code>a := 5; b := 3;</code> <code>write('Z(' , a, ')=(, a+b, ')');</code> |
4. Запишите оператор для вывода значений целых переменных `a=5` и `b=3` в формате:
- | | |
|----|---------------------------|
| а) | <code>3+5=?</code> |
| б) | <code>Z(5)=F(3)</code> |
| в) | <code>a=5; b=3;</code> |
| г) | Ответ: <code>(5;3)</code> |

§ 56. Вычисления

Контрольные вопросы

- Какие типы данных вы знаете?
- Как вы думаете, почему во многих языках программирования есть несколько целочисленных и вещественных типов данных?
- Чем отличается символьная переменная от строковой?
- Какие данные записываются в логические переменные? Как вы думаете, сколько места в памяти они занимают?
- Что такое приоритет операций? Зачем он нужен?
- В каком порядке выполняются операции, если они имеют одинаковый приоритет?
- Зачем используются скобки?
- Что происходит, если в выражения входят переменные разных типов? Какого типа будет результат?
- Опишите операции `div` и `mod`. Подумайте, почему они не определены для вещественных чисел.
- Расскажите о проблеме вычисления остатка в различных языках программирования. Обсудите в классе этот вопрос.
- Как выполняется операция возведения в степень?
- Какие стандартные математические функции вы знаете? В каких единицах задается аргумент тригонометрических функций?
- Как выполнить округление (к ближайшему целому) в алгоритмическом языке?
- Какие числа называют случайными? Зачем они нужны?
- Как получить «естественное» случайное число? Почему такие числа почти не используются в цифровой технике?
- Чем отличаются псевдослучайные числа от случайных?
- Какие функции для получения псевдослучайных чисел вы знаете?

Задачи и задания

- Найдите в справочной системе или в Интернете диапазон значений для вещественных типов данных.
- Напишите программу, которая находит сумму, произведение и среднее арифметическое трёх целых чисел, введённых с клавиатуры. Например, при вводе чисел 4, 5 и 7 мы должны получить ответ
`4+5+7=16, 4*5*7=140, (4+5+7)/3=5.333333`
- Напишите программу, которая вводит радиус круга и вычисляет его площадь и длину окружности. На языке Паскаль можно использовать встроенную константу `Pi`, равную числу π .
- Напишите программу, которая меняет местами значения двух переменных в памяти.

5. *В предыдущей задаче попробуйте найти решение, которое не использует дополнительные переменные.
6. Напишите программу, которая возводит введенное число в степень 10, используя только операции сложения и умножения. Что произойдет, если ввести большое число, например, 78? Попробуйте объяснить полученный результат.
7. Вычислите значение вещественной переменной c при $a = 2$ и $b = 3$:
- | | | |
|----|---|--|
| а) | <code>c := a + 1 / 3</code> | |
| б) | <code>c := a + 4 / 2 * 3 + 6</code> | |
| в) | <code>c := (a + 4) / 2 * 3</code> | |
| г) | <code>c := (a + 4) / (b + 3) * a</code> | |
8. Вычислите значение целочисленной переменной c при $a = 26$ и $b = 6$:
- | | | |
|----|--|--|
| а) | <code>c := mod(a, b) + b</code> | <code>c := a mod b + b;</code> |
| б) | <code>c := div(a, b) + a</code> | <code>c := a div b + a;</code> |
| в) | <code>b := div(a, b)</code> <code>c := div(a, b)</code> | <code>b := a div b;</code> <code>c := a div b;</code> |
| г) | <code>b := div(a, b) + b</code> <code>c := mod(a, b) + a</code> | <code>b := a div b + b;</code> <code>c := a mod b + a;</code> |
| д) | <code>b := mod(a, b) + 4</code> <code>c := mod(a, b) + 1</code> | <code>b := a mod b + 4;</code> <code>c := a mod b + 1;</code> |
| е) | <code>b := div(a, b)</code> <code>c := mod(a, b+1)</code> | <code>b := a div b;</code> <code>c := a mod (b+1);</code> |
| ж) | <code>b := mod(a, b)</code> <code>c := div(a, b+1)</code> | <code>b := a mod b;</code> <code>c := a div (b+1);</code> |
9. Выполните предыдущее задание при $a = -22$ и $b = 4$. Чем отличаются результаты работы программ на алгоритмическом языке и на Паскале?
10. Напишите программу, которая вводит трёхзначное число и разбивает его на цифры. например, при вводе числа 123 программа должна вывести «1,2,3».
11. Напишите программу, которая вводит координаты двух точек на числовой оси и выводит расстояние между ними. Учитывайте, что первой может быть введена меньшая координата.
12. Напишите программы на обоих языках, которые вводят два вещественных числа (x и y) и вычисляют значение x^y .
13. Напишите программу на алгоритмическом языке, которая округляет вещественное число до ближайшего целого.
14. Напишите программу, которая вводит два целых числа, a и b ($a < b$), и выводит на экран 5 случайных целых чисел на отрезке $[a, b]$.
15. Напишите программу, которая моделирует бросание двух игральных кубиков: при запуске выводит случайное число в диапазоне от 2 до 12.
16. Напишите программу, которая случайным образом выбирает дежурных: выводит два случайных числа в диапазоне от 1 до N , где N – количество учеников вашего класса. Какая проблема может при этом возникнуть?
17. Напишите программу, которая вводит два вещественных числа, a и b ($a < b$), и выводит на экран 5 случайных вещественных чисел в полуинтервале $[a, b)$.

§ 57. Ветвления

Контрольные вопросы

1. Чем отличаются разветвляющиеся алгоритмы от линейных?
2. Как вы думаете, почему не все задачи можно решить с помощью линейных алгоритмов?
3. Как вы думаете, хватит ли линейных алгоритмов и ветвлений для разработки любой программы?

4. Почему нельзя выполнить обмен значений двух переменных в два шага: $a := b$; $b := a$?
5. Чем отличаются условные операторы в полной и неполной формах? Как вы думаете, можно ли обойтись только неполной формой?
6. Какие отношения вы знаете? Как обозначаются отношения «равно» и «не равно»?
7. Что такое сложное условие?
8. Как определяется порядок вычислений в сложном условии? Расскажите об особенностях вычисления логических выражений в языке Паскаль.
9. Зачем нужен оператор выбора? Как можно обойтись без него?
10. Расскажите о различиях в операторах выбора в алгоритмическом языке и в Паскале.
11. Как в операторе выбора определить, что нужно делать, если ни один вариант не подошёл?
12. Как в операторе выбора в языке Паскаль выполнить для какого-то варианта несколько операторов?



Задачи и задания

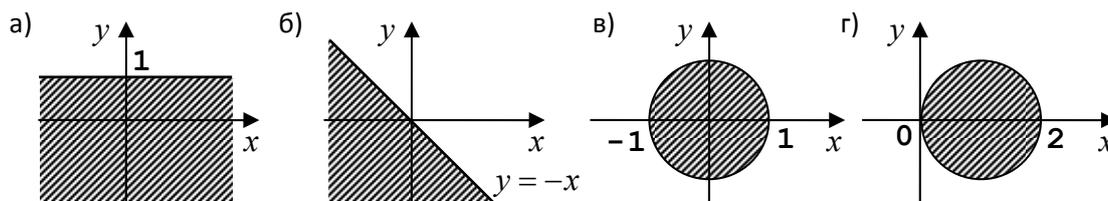
1. Покажите, что приведенная программа не всегда верно определяет максимальное из трёх чисел, записанных в переменные a , b и c :

```
если a > b то M := a
иначе      M := b все
если c > b то M := c
иначе      M := b все
```

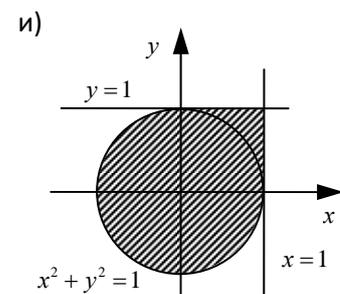
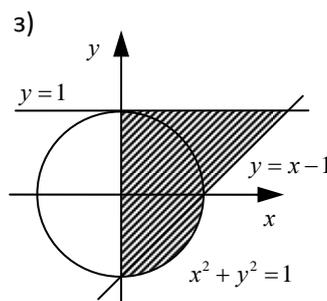
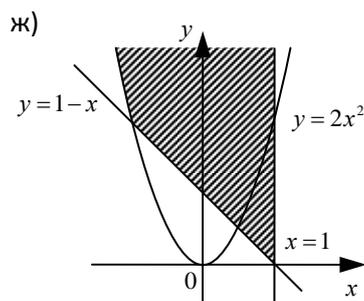
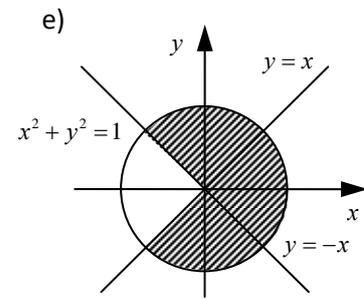
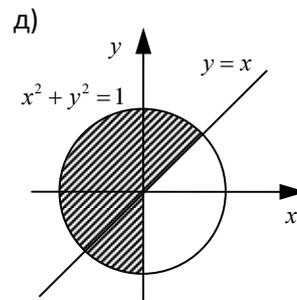
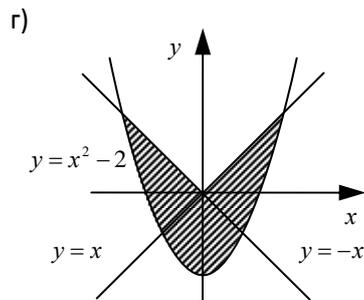
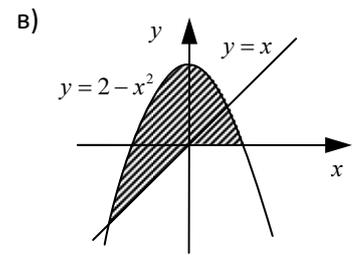
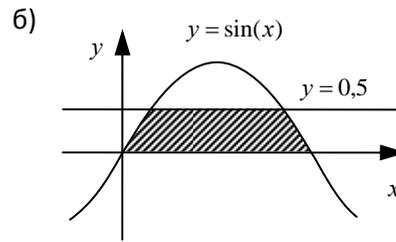
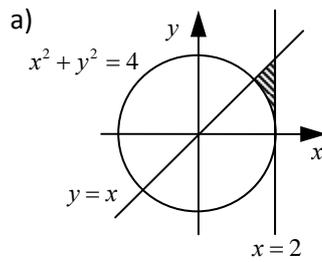
```
if a > b then M := a
else        M := b;
if c > b then M := c
else        M := b;
```

Приведите контрпример, то есть значения переменных, при котором в переменной M будет получен неверный ответ. Как нужно доработать программу, чтобы она всегда работала правильно?

2. Напишите программу, которая выбирает максимальное и минимальное из пяти введённых чисел.
3. Напишите программу, которая определяет, верно ли, что введённое число – трёхзначное.
4. Напишите программу, которая вводит номер месяца и выводит название времени года. Оператор выбора использовать не разрешается. При вводе неверного номера месяца должно быть выведено сообщение об ошибке.
5. Решите предыдущую задачу с помощью оператора выбора.
6. Напишите программу, которая вводит с клавиатуры номер месяца и определяет, сколько дней в этом месяце. При вводе неверного номера месяца должно быть выведено сообщение об ошибке.
7. Напишите программу, которая вводит с клавиатуры номер месяца и день, и определяет, сколько дней осталось до Нового года. При вводе неверных данных должно быть выведено сообщение об ошибке.
8. Напишите программу, которая вводит возраст человека (целое число, не превышающее 120) и выводит этот возраст со словом «год», «года» или «лет». Например, «21 год», «22 года», «25 лет».
9. Напишите программу, которая вводит целое число, не превышающее 100, и выводит его прописью, например, 21 → «двадцать один».
10. Напишите программу, которая вводит координаты точки на плоскости и определяет, попала ли эта точка в заштрихованную область.



11. Напишите два варианта программы, которая вводит координаты точки на плоскости и определяет, попала ли эта точка в заштрихованную область. Один вариант программы должен использовать сложные условия, второй – обходиться без них.



§ 58. Циклические алгоритмы



Контрольные вопросы

1. Что такое цикл?
2. Сравните цикл с переменной и цикл с условием. Какие преимущества и недостатки есть у каждого из них?
3. Что означает выражение «цикл с предусловием»?
4. В каком случае цикл с предусловием не выполняется ни разу?
5. В каком случае программа, содержащая цикл с условием, может зациклиться?
6. В каком случае цикл с переменной не выполняется ни разу?
7. Верно ли, что любой цикл с переменной можно заменить циклом с условием? Верно ли обратное утверждение?
8. В каком случае можно заменить цикл с условием на цикл с переменной?
9. Как будет работать приведенная программа, которая считает количество цифр введённого числа, при вводе отрицательного числа? Если вы считаете, что она работает неправильно, укажите, как её нужно доработать.



Задачи и задания

1. Найдите ошибку в программе:

```
k := 0
нц пока k < 10
```

```
k := 0;
while k < 10 do
```

```
вывод 'привет', нс
кц
```

```
writeln('привет');
```

Как её можно исправить?

2. Напишите программу, которая вводит два целых числа и находит их произведение, не используя операцию умножения. Учтите, что числа могут быть отрицательными.
3. Напишите программу, которая вводит натуральное число **N** и находит сумму всех натуральных чисел от 1 до **N**. Используйте сначала цикл с условием, а потом – цикл с переменной.
4. Напишите программу, которая вводит натуральное число **N** и выводит первые **N** чётных натуральных чисел.
5. Напишите программу, которая вводит натуральные числа **a** и **b**, и выводит квадраты натуральных чисел в интервале от **a** до **b**. Например, если ввести 4 и 5, программа должна вывести

$$4 * 4 = 16$$

$$5 * 5 = 25$$
6. Напишите программу, которая вводит натуральные числа **a** и **b**, и выводит сумму квадратов натуральных чисел в интервале от **a** до **b**.
7. Напишите программу, которая вводит натуральное число **N** и выводит на экран **N** псевдослучайных чисел. Запустите её несколько раз, объясните результаты опыта.
8. Напишите программу, которая строит последовательность из **N** случайных чисел на отрезке от 0 до 1 и определяет, сколько из них попадает в полуинтервалы [0; 0,25), [0,25; 0,5), [0,5; 0,75) и [0,75; 1). Сравните результаты, полученные при **N** = 10, 100, 1000, 10000.
9. Найдите все пятизначные числа, которые при делении на 133 дают в остатке 125, а при делении на 134 дают в остатке 111.
10. Напишите программу, которая вводит натуральное число **N** и выводит на экран все натуральные числа, не превосходящие **N** и делящиеся на каждую из своих цифр.
11. *Числа Армстронга.* Натуральное число называется числом Армстронга, если сумма цифр числа, возведенных в **N**-ную степень (где **N** – количество цифр в числе) равна самому числу. Например, $153 = 1^3 + 5^3 + 3^3$. Найдите все трёхзначные и четырёхзначные числа Армстронга.
12. *Аutomорфные числа.* Натуральное число называется автоморфным, если оно равно последним цифрам своего квадрата. Например, $25^2 = 625$. Напишите программу, которая вводит натуральное число **N** и выводит на экран все автоморфные числа, не превосходящие **N**.
13. Напишите программу, которая считает количество чётных цифр введённого числа.
14. Напишите программу, которая считает сумму цифр введённого числа.
15. Напишите программу, которая определяет, верно ли, что введённое число содержит две одинаковых цифры, стоящие рядом (как, например, 221).
16. Напишите программу, которая определяет, верно ли, что введённое число состоит из одинаковых цифр (как, например, 222).
17. *Напишите программу, которая определяет, верно ли, что введённое число содержит по крайней мере две одинаковых цифры, возможно, не стоящие рядом (как, например, 212).
18. Используя сначала цикл с условием, а потом – цикл с переменной, напишите программу, которая выводит на экран чётные степени числа 2 от 2^{10} до 2^2 в порядке убывания.
19. *Алгоритм Евклида* для вычисления наибольшего общего делителя двух натуральных чисел формулируется так: нужно заменять большее число на разность большего и меньшего до тех пор, пока одно из них не станет равно нулю; тогда второе и есть НОД. Напишите программу, которая реализует этот алгоритм. Какой цикл тут нужно использовать?
20. Напишите программу, использующую *модифицированный алгоритм Евклида*: нужно заменять большее число на остаток от деления большего на меньшее до тех пор, пока этот остаток не станет равен нулю; тогда второе и есть НОД.
21. Добавьте в решение двух предыдущих задач вычисление количества шагов цикла. Заполните таблицу (шаги-1 и шаги-2 означают количество шагов двух версия алгоритма Евклида):

| | | | | | |
|-------------------|--------------|---------------|-----------------|-----------------|------------------|
| a | 64168 | 358853 | 6365133 | 17905514 | 549868978 |
| b | 82678 | 691042 | 11494962 | 23108855 | 298294835 |
| НОД (a, b) | | | | | |
| шаги-1 | | | | | |
| шаги-2 | | | | | |

22. Напишите программу, которая вводит с клавиатуры 10 чисел и вычисляет их сумму и произведение.
23. Напишите программу, которая вводит с клавиатуры числа до тех пор, пока не будет введено число 0. В конце работы программы на экран выводится сумма и произведение введенных чисел.
24. Напишите программу, которая вводит с клавиатуры числа до тех пор, пока не будет введено число 0. В конце работы программы на экран выводится минимальное и максимальное из введенных чисел.
25. Напишите программу, которая вводит с клавиатуры натуральное число **N** и определяет его *факториал*, то есть произведение натуральных чисел от 1 до **N**: $N! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot N$. Что будет, если ввести большое значение **N** (например, 20)?
26. Напишите программу, которая вводит натуральные числа **A** и **N** и вычисляет A^N .
27. Напишите программу, которая выводит на экран все цифры числа, начиная с первой.
28. Ряд чисел Фибоначчи задается следующим образом: первые два числа равны 1 ($F_1 = F_2 = 1$), а каждое следующее равно сумме двух предыдущих: $F_n = F_{n-1} + F_{n-2}$. Напишите программу, которая вводит натуральное число **N** и выводит на экран первые **N** чисел Фибоначчи.
29. Напишите программу, которая вводит натуральные числа **a** и **b** и выводит все простые числа в диапазоне от **a** до **b**.
30. Совершенным называется число, равное сумме всех своих делителей, меньших его самого (например, число $6 = 1 + 2 + 3$). Напишите программу, которая вводит натуральное число **N** и определяет, является ли число **N** совершенным.
31. Напишите программу, которая вводит натуральное число **N** и находит все совершенные числа в диапазоне от 1 до **N**.
32. В магазине продается мастика в ящиках по 15 кг, 17 кг, 21 кг. Как купить ровно 185 кг мастики, не вскрывая ящики? Сколькими способами можно это сделать?
33. *Ввести натуральное число **N** и вывести значение числа $1/N$, выделив период дроби. Например, $1/2 = 0,5$ или $1/7 = 0,(142857)$.
34. *В телевикторине участнику предлагают выбрать один из трёх закрытых чёрных ящиков, причём известно, что в одном из них – приз, а в двух других – пусто. После этого ведущий открывает один пустой ящик (но не тот, который выбрал участник) и предлагает заново сделать выбор, но уже между двумя оставшимися ящиками. Используя псевдослучайные числа, выполните моделирование 1000 раундов этой игры и определите, что выгоднее делать участнику викторины: выбрать тот же ящик, что и в начале игры, или другой¹.

§ 59. Процедуры

Контрольные вопросы

1. Что такое процедуры? В чем смысл их использования?
2. Как оформляются процедуры в алгоритмическом языке и в Паскале?
3. Достаточно ли включить процедуру в текст программы, чтобы она «сработала»?

¹ Эта задача известна как парадокс Монти Холла, потому что её решение противоречит интуиции и «здравому смыслу».

4. Что такое параметры? Зачем они используются?
5. Какие переменные называются локальными? Где они объявляются?
6. Как оформляются процедуры, имеющие несколько параметров?
7. Что такое изменяемые параметры? Зачем они используются?
8. Как в заголовке процедуры отличить изменяемый параметр от неизменяемого?
9. Какие типы параметров выделяются в алгоритмическом языке? Как они объявляются?



Задачи и задания

1. Напишите процедуру, которая выводит на экран в столбик все цифры переданного ей числа, начиная с последней.
2. Напишите процедуру, которая выводит на экран в столбик все цифры переданного ей числа, начиная с первой.
3. Напишите процедуру, которая выводит на экран все делители переданного ей числа (в одну строчку).
4. *Напишите процедуру, которая выводит на экран запись переданного ей числа в римской системе счисления.
5. Напишите процедуру, которая выводит на экран запись числа, меньшего, чем 8^{10} , в виде 10 знаков в восьмеричной системе счисления.
6. Напишите процедуру, которая выводит на экран запись числа, меньшего, чем $16^4 = 65536$, в виде 4-х знаков в шестнадцатеричной системе счисления.
7. Напишите процедуру, которая принимает параметр – натуральное число N – и выводит на экран линию из N символов '- '.
8. Напишите процедуру, которая принимает параметр – натуральное число N – и выводит на экран квадрат из звездочек со стороной N .
9. Напишите процедуру, которая принимает числовой параметр – возраст человека в годах, и выводит этот возраст со словом «год», «года» или «лет». Например, «21 год», «22 года», «12 лет».
10. Напишите процедуру, которая выводит переданное ей число прописью. Например, 21 → «двадцать один».
11. Напишите процедуру, которая принимает параметр – натуральное число N – и выводит первые N чисел Фибоначчи (см. задания к § 58.).
12. Напишите процедуру, которая определяет, верно ли, что переданное ей число – простое. (Используйте изменяемые параметры).

§ 60. Функции



Контрольные вопросы

1. Что такое функция? Чем она отличается от процедуры?
2. Как оформляются функции в тексте программы (сравните алгоритмический язык и Паскаль)?
3. Как по тексту программы определить, какое значение возвращает функция?
4. Какие функции называются логическими? Зачем они нужны?



Задачи и задания

1. Напишите функцию, которая вычисляет максимальное из трёх чисел.
2. Напишите функцию, которая вычисляет сразу максимальное и минимальное из трёх чисел. (Используйте изменяемые параметры.)
3. Напишите функцию, которая вычисляет количество цифр числа.
4. Напишите функцию, которая вычисляет наибольший общий делитель двух чисел.
5. Напишите функцию, которая вычисляет наименьшее общее кратное двух чисел.
6. Напишите функцию, которая «разворачивает» десятичную запись числа наоборот, например, из 123 получается 321, и из 3210 – 0123.

7. Напишите функцию, которая моделирует бросание двух игральных кубиков (на каждом может выпасть от 1 до 6 очков). (Используйте генератор псевдослучайных чисел.)
8. Напишите функцию, которая вычисляет факториал натурального числа N .
9. Напишите функцию, которая вычисляет N -ое число Фибоначчи.
10. *Дружественные числа* – это два натуральных числа, таких, что сумма всех делителей одного числа (меньших самого этого числа) равна другому числу, и наоборот. Найдите все пары дружественных чисел, каждое из которых меньше 10000. Используйте функцию, которая вычисляет сумму делителей числа.
11. Напишите программу, которая вводит натуральное число N и находит все числа в диапазоне $[0, N]$, сумма цифр которых не меняется при умножении на 2, 3, 4, 5, 6, 7, 8 и 9 (например, число 9). Используйте функцию для вычисления суммы цифр числа.
12. Напишите логическую функцию, которая определяет, верно ли, что число N – совершенное, то есть равно сумме своих делителей, меньших его самого.
13. Простое число называется гиперпростым, если любое число, получающееся из него откидыванием нескольких цифр, тоже является простым. Например, число 733 – гиперпростое, так как и оно само, и числа 73 и 7 – простые. Напишите логическую функцию, которая определяет, верно ли, что число N – гиперпростое. Используйте уже готовую функцию `isPrime`.

§ 61. Рекурсия



Контрольные вопросы

1. Что такое рекурсия? Приведите примеры.
2. Как вы думаете, почему любое рекурсивное определение состоит из двух частей?
3. Что такое рекурсивная процедура (функция)?
4. Расскажите о задаче «Ханойские башни». Попытайтесь придумать алгоритм ее решения, не использующий рекурсию.
5. Процедура А вызывает процедуру Б, а процедура Б – процедуру А и сама себя. Какую из них можно назвать рекурсивной?
6. В каком случае рекурсия никогда не остановится? Докажите, что в рассмотренных задачах этого не случится.
7. Что такое стек? Как он используется при выполнении программ?
8. Почему при использовании рекурсии может случиться переполнение стека?
9. Назовите достоинства и недостатки рекурсии. Когда ее следует использовать, а когда – нет?



Задачи и задания

1. Найдите в Интернете информацию об использовании рекурсии в искусстве и рекламе. Сделайте сообщение в классе.
2. Найдите в Интернете информацию о фракталах. Сделайте сообщение в классе.
3. Используя материалы Интернета, ответьте на вопрос: «Как связаны числа Фибоначчи с кроликами?»
4. Придумайте свою рекурсивную фигуру и опишите её.
5. *Используя графические возможности вашего языка программирования, постройте на экране треугольник Серпинского и другие фракталы.
6. Напишите рекурсивную процедуру для перевода числа в двоичную систему, которая правильно работала бы для нуля (выводила 0).
7. *Напишите рекурсивную процедуру для перевода числа в шестнадцатеричную систему счисления.
8. *Напишите рекурсивную процедуру для перевода числа в троичную уравновешенную систему счисления (см. § 14). Вместо цифры $\bar{1}$ используйте символ «#». *Дано натуральное число N . Требуется получить и вывести на экран все возможные *различные* способы представления этого числа в виде суммы натуральных чисел (то есть, $1 + 2$ и $2 + 1$ – это один и тот же способ разложения числа 3). Решите задачу с помощью рекурсивной процедуры

9. Напишите рекурсивную процедуру для перевода числа из двоичной системы счисления в десятичную.
10. *Напишите рекурсивную процедуру для перевода числа из шестнадцатеричной системы счисления в десятичную.*Напишите рекурсивную процедуру для перевода числа из троичной уравновешенной системы счисления (см. § 14) в десятичную. Вместо цифры $\bar{1}$ используйте символ «#».
11. Напишите рекурсивную и нерекурсивную функции, вычисляющие НОД двух натуральных чисел с помощью модифицированного алгоритма Евклида. Какой вариант вы предпочтете?

§ 62. Массивы



Контрольные вопросы

1. Что такое массив? Зачем нужны массивы?
2. Зачем нужно объявлять массивы?
3. Как объявляются массивы в алгоритмическом языке и в Паскале?
4. Как вы думаете, почему элементы массива расположены в памяти рядом?
5. Как обращаются к элементу массива?
6. Могут ли индексы элементов массива начинаться с 0? с -5?
7. Почему размер массива лучше вводить как константу, а не число?
8. Как ввести массив и вывести его на экран?
9. Как заполнить массив случайными числами в диапазоне от 100 до 200?



Задачи и задания

1. Заполните массив элементами арифметической прогрессии. Её первый элемент и разность нужно ввести с клавиатуры.
2. Заполните массив степенями числа 2 (от 2^1 до 2^N).
3. Заполните массив первыми числами Фибоначчи.
4. *Заполните массив из N элементов случайными целыми числами в диапазоне 1..N так, чтобы в массив обязательно вошли все числа от 1 до N (постройте случайную перестановку).
5. *Постройте случайную перестановку чисел от 1 до N так, чтобы первое число обязательно было равно 5.
6. Заполните массив случайными числами в диапазоне 20..100 и подсчитайте отдельно число чётных и нечётных элементов.
7. Заполните массив случайными числами в диапазоне 1000..2000 и подсчитайте число элементов, у которых вторая с конца цифра – чётная.
8. Заполните массив случайными числами в диапазоне 0..100 и подсчитайте отдельно среднее значение всех элементов, которые <50 , и среднее значение всех элементов, которые ≥ 50 .

§ 63. Алгоритмы обработки массивов



Контрольные вопросы

1. Почему при поиске индекса максимального элемента не нужно хранить само значение максимального элемента?
2. Что такое реверс массива?
3. Как вы думаете, какую ошибку чаще всего делают начинающие, программируя реверс массива?
4. Как вы думаете, какие проблемы (и ошибки) могут возникнуть при циклическом сдвиге массива *вправо*?
5. Что произойдет с массивом при выполнении следующего фрагмента программы:

```
нц для i от 1 до N-1
  A[i+1] := A[i]
кц
```

```
for i:=1 to N-1 do
  A[i+1] := A[i];
```

6. Как (при использовании приведенного алгоритма поиска) определить, что элемент не найден?
7. Что такое выход за границы массива? Почему он может быть опасен?
8. Опишите «очевидный» алгоритм отбора части элементов одного массива в другой массив. Почему его не используют?



Задачи и задания

1. Напишите программу, которая находит максимальный и минимальный из чётных положительных элементов массива. Если в массиве нет чётных положительных элементов, нужно вывести сообщение об этом.
2. Введите массив с клавиатуры и найдите (за один проход) количество элементов, имеющих максимальное значение.
3. Найдите за один проход по массиву три его различных элемента, которые меньше всех остальных («три минимума»).
4. *Заполните массив случайными числами в диапазоне 10..12 и найдите длину самой длинной последовательности стоящих рядом одинаковых элементов.
5. Заполните массив случайными числами в диапазоне 0..4 и выведите на экран номера всех элементов, равных значению **X** (оно вводится с клавиатуры).
6. Заполните массив случайными числами и переставьте соседние элементы, поменяв 1-ый элемент со 2-м, 3-й – с 4-м и т.д.
7. Заполните массив из чётного количества элементов случайными числами и выполните реверс отдельно для 1-ой и 2-ой половин массива.
8. Заполните массив случайными числами и выполните реверс для части массива между элементами с индексами **K** и **M** (включая эти элементы).
9. Напишите программу для выполнения циклического сдвига массива вправо на 4 элемента.
10. Найдите в массиве все простые числа и скопируйте их в новый массив.
11. *Найдите в массиве все числа Фибоначчи и скопируйте их в новый массив.

§ 64. Сортировка



Контрольные вопросы

1. Что такое сортировка?
2. На какой идее основан метод пузырька? метод выбора?
3. Объясните, зачем нужен вложенный цикл в описанных методах сортировки.
4. Сравните метод пузырька и метод выбора. Какой из них требует меньше перестановок?
5. Расскажите про основные идеи метода «быстрой сортировки».
6. Как вы думаете, можно ли использовать метод «быстрой сортировки» для нечисловых данных, например, для символьных строк?
7. От чего зависит скорость «быстрой сортировки»? Какой самый лучший и самый худший случай?
8. Как вы думаете, может ли метод «быстрой сортировки» работать дольше, чем метод выбора (или другой «простой» метод)? Если да, то при каких условиях?
9. Как нужно изменить приведенные алгоритмы, чтобы элементы массива были отсортированы по убыванию?



Задачи и задания

1. Отсортировать массив и найти количество различных чисел в нём.
2. Напишите программу, в которой сортировка выполняется «методом камня» – самый тяжёлый элемент опускается в конец массива.
3. Напишите программу, которая выполняет неполную сортировку массива: ставит в начало массива три самых меньших по величине элемента в порядке возрастания (неубывания).

4. Напишите вариант метода пузырька, который заканчивает работу, если на очередном шаге внешнего цикла не было перестановок.
5. Напишите программу, которая сортирует массив по возрастанию последней цифры числа.
6. Напишите программу, которая сортирует массив по убыванию суммы цифр числа.
7. Напишите программу, которая сортирует первую половину массива по возрастанию, а вторую – по убыванию (элементы из первой половины не должны попадать во вторую и наоборот).
8. Напишите программу, которая сортирует массив, а затем находит максимальное из чисел, встречающихся в массиве несколько раз.
9. *Напишите программу, которая сравнивает количество перестановок при сортировке одного и того же массива разными методами. Проведите эксперименты для возрастающей последовательности (уже отсортированной), убывающей (отсортированной в обратном порядке) и случайной.

§ 65. Двоичный поиск

Контрольные вопросы

1. Почему этот алгоритм поиска называется «двоичным»?
2. Приведите примеры использования двоичного поиска в обычной жизни.
3. Как можно примерно подсчитать количество шагов при двоичном поиске?
4. Сравните достоинства и недостатки линейного и двоичного поиска.

Задачи и задания

1. Напишите программу, которая сортирует массив по убыванию и ищет в нем все значения, равные введенному числу.
2. Напишите программу, которая считает среднее число шагов при двоичном поиске для массива из 32 элементов в диапазоне 0..100. Для поиска используйте 1000 случайных чисел в этом же диапазоне.

§ 66. Символьные строки

Контрольные вопросы

1. Что такое символьная строка?
2. Почему неудобно заменять строки массивами символов?
3. Как объявляются строки в алгоритмическом языке и в Паскале?
4. Как обращаться к элементу строки с заданным номером?
5. Как вычисляется длина строки?
6. Что обозначает оператор ' + ' применительно к строкам?
7. Перечислите основные операции со строками и соответствующие им стандартные функции.
8. Как определить, что при поиске в строке образец не найден?
9. Чем отличаются средства алгоритмического языка и Паскаля для работы со строками?
10. Как преобразовать число из символьного вида в числовой и обратно?
11. Почему строку не всегда можно преобразовать в число? Как определить, что преобразование закончилось неудачно?
12. Объясните выражение «рекурсивный перебор».
13. Сравните рекурсивные и нерекурсивные методы решения переборных задач.

Задачи и задания

1. Напишите программу, которая во введенной символьной строке заменяет все буквы «а» на буквы «б» и наоборот, как заглавные, так и строчные. При вводе строки ' **абсАБС** ' должен получиться результат ' **басБАС** '.

2. Ввести символьную строку и проверить, является ли она *палиндромом* (палиндром читается одинаково в обоих направлениях, например, «казак»).
3. Ввести адрес файла и «разобрать» его на части, разделенные знаком '/'. Каждую часть вывести в отдельной строке.
4. Ввести строку, в которой записана сумма натуральных чисел, например, '1+25+3'. Вычислите это выражение.
5. Ввести с клавиатуры в одну строку фамилию, имя и отчество, разделив их пробелом. Вывести фамилию и инициалы. Например, при вводе строки 'Иванов Петр Семёнович' должно получиться 'П.С. Иванов'.
6. Разберитесь, как работает еще одна функция замены:

```

алг лит ReplaceBad(лит s0, sOld, sNew)
нач
  лит s
  s := s0
  цел p, len
  len := длин(sOld)
  p := позиция(sOld, s)
  нц пока p > 0
    удалить(s, p, len)
    вставить(sNew, s, p)
    p := позиция(sOld, s)
  кц
  знач := s
кон

```

Приведите пример входных данных, при которых эта функция работает неправильно.

7. Напишите рекурсивную версию процедуры `replaceAll`. Сравните две версии. Какую из них вы рекомендуете выбрать и почему?
8. Напишите функцию, которая изменяет в имени файла расширение на заданное (например, на '.bak'). Функция принимает два параметра: имя файла и нужно расширение. Учтите, что в исходном имени расширение может быть пустым.
9. Напишите функцию, которая определяет, сколько раз входит в символьную строку заданное слово.
10. С клавиатуры вводится число **N**, обозначающее количество футболистов команды «Бублик», а затем – **N** строк, в каждой из которых – информация об одном футболисте таком формате:


```
<Фамилия> <Имя> <год рождения> <голы>
```

 Все данные разделяются одним пробелом. Нужно подсчитать, сколько футболистов, родившихся в период с 1998 по 2000 год, не забивали мячей вообще.
11. В условиях предыдущей задачи определите фамилию и имя футболиста, забившего наибольшее число голов, и количество забитых им голов.
12. В условиях предыдущей задачи выведите в алфавитном порядке фамилии и имена всех футболистов, которые забивали хотя бы один гол. В списке не более 100 футболистов.
13. Измените программу рекурсивного перебора так, чтобы длину слова можно было ввести с клавиатуры.
14. Выведите на экран все слова из **K** букв, в которых буква «Ы» встречается более 1 раза, и подсчитайте их количество.
15. Выведите на экран все слова из **K** букв, в которых есть одинаковые буквы, стоящие рядом (например, «ЫШШО»), и подсчитайте их количество.
16. В языке племени «тумба-юмба» запрещено ставить две гласные буквы подряд. Выведите все слова длины **K**, удовлетворяющие этому условию, и найдите их количество.
17. *Напишите программу перебора слов заданной длины, не использующую рекурсию. Попробуйте составить функцию, которая на основе некоторой комбинации вычисляет следующую за ней.
18. *Перестановки. К вам пришли **K** гостей. Напишите программу, которая выводит все *перестановки* – способы посадить их за столом. Гости можно обозначить латинскими буквами.

§ 67. Матрицы



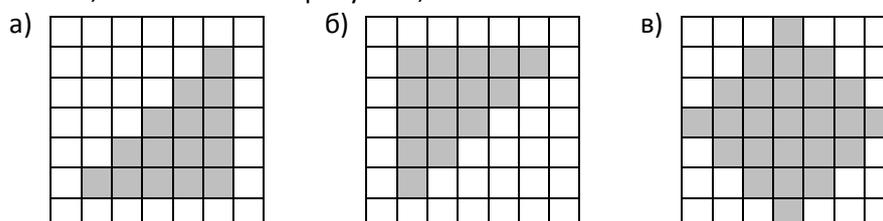
Контрольные вопросы

1. Что такое матрицы? Зачем они нужны?
2. Сравните понятия «массив» и «матрица».
3. Как вы думаете, можно ли считать, что первый индекс элемента матрицы – это номер столбца, а второй – номер строки?
4. Могут ли индексы элементов матрицы принимать отрицательные и нулевые значения?
5. Что такое главная и побочная диагонали матрицы?
6. Почему суммирование элементов главной диагонали требует одиночного цикла, а суммирование элементов под главной диагональю – вложенного?

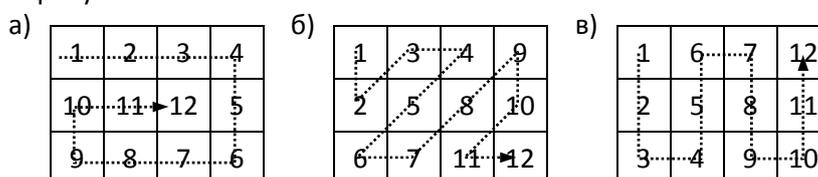


Задачи и задания

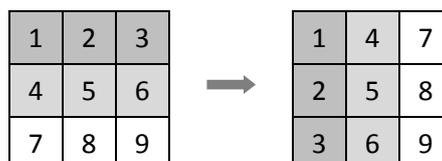
1. Напишите программу, которая находит минимальный и максимальный элементы матрицы и их индексы.
2. Напишите программу, которая находит минимальный и максимальный из чётных положительных элементов матрицы и их индексы. Учтите, что таких элементов в матрице может и не быть.
3. Напишите программу, которая выводит на экран строку матрицы, сумма элементов которой наибольшая.
4. Напишите программу, которая выводит на экран столбец матрицы, сумма элементов которой наименьшая.
5. Напишите программу, которая заполняет матрицу случайными числами, а затем записывает нули во все элементы выше главной диагонали.
6. Напишите программу, которая заполняет матрицу случайными числами, а затем записывает нули во все элементы выше побочной диагонали.
7. Напишите программу, которая заполняет матрицу 7×7 случайными числами, а затем записывает в элементы, отмеченные на рисунках, число 99:



8. Заполните матрицу, содержащую N строк и M столбцов, натуральными числами по спирали и змейкой, как на рисунках:



9. Заполните квадратную матрицу случайными числами и выполните её *транспонирование*: так называется процедура, в результате которой строки матрицы становятся столбцами, а столбцы – строками:



10. Заполните квадратную матрицу случайными числами и выполните её поворот на 90 градусов:

| | | |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | 9 |

→

| | | |
|---|---|---|
| 7 | 4 | 1 |
| 8 | 5 | 2 |
| 9 | 6 | 3 |

11. *Напишите программу, которая играет с человеком в крестики-нолики.
12. *В матрице, содержащей **N** строк и **M** столбцов, записана карта островного государства Лимония (нули обозначают море, а единицы – сушу). Все острова имеют форму прямоугольника. Написать программу, которая по готовой карте определяет количество островов.

§ 68. Работа с файлами

Контрольные вопросы

1. Чем отличаются текстовые и двоичные файлы по внутреннему содержанию? Можно ли сказать, что текстовый файл – это частный случай двоичного файла?
2. Объясните «принцип сэндвича» при работе с файлами.
3. Как вы думаете, почему открытый программой файл, как правило, блокируется и другие программы не могут получить к нему доступ?
4. Почему рекомендуется вручную закрывать файлы, хотя при закрытии программы они закроются автоматически? В каких ситуациях это может быть важно?
5. Что такое файловая переменная? Почему для работы с файлом используют не имя файла, а файловую переменную?
6. В каком случае одна и та же файловая переменная может быть использована для работы с несколькими файлами, а в каком – нет?
7. Что такое «последовательный доступ к данным»?
8. Как можно начать чтение данных из файла с самого начала?
9. Как определить, что данные в файле закончились?
10. В каких случаях нужно знать максимальное количество данных в файле, а в каких – нет?
11. В каких случаях нужно открывать одновременно несколько файлов?

Задачи и задания

1. Напишите программу, которая находит среднее арифметическое всех чисел, записанных в файле в столбик, и выводит результат в другой файл.
2. Напишите программу, которая находит минимальное и максимальное среди чётных положительных чисел, записанных в файле, и выводит результат в другой файл. Учтите, что таких чисел может вообще не быть.
3. В файле в столбик записаны целые числа. Напишите программу, которая определяет длину самой длинной цепочки идущих подряд одинаковых чисел и выводит результат в другой файл.
4. В файле записано не более 100 чисел. Отсортировать их по возрастанию последней цифры и записать в другой файл.
5. В файле записано не более 100 чисел. Отсортировать их по возрастанию суммы цифр и записать в другой файл.
6. В двух файлах записаны отсортированные по возрастанию массивы неизвестной длины. Объединить их и записать результат в третий файл. Полученный массив также должен быть отсортирован по возрастанию.
7. Дополните решение задачи о собаках, так чтобы программа обрабатывала ошибки в исходных данных. При любых ошибках программа не должна завершаться аварийно.
8. В исходном файле записана речь подростка, в которой часто встречается слово-паразит «короче», например: «Мама, короче, мыла, короче, раму.» Убрать из текста все слова-паразиты (должно остаться «Мама мыла раму.»).
9. Прочитать текст из файла и подсчитать количество слов в нём.

10. Прочитать текст из файла и вывести в другой файл только те строки, в которых есть слова, начинающиеся с буквы А.
11. Прочитать текст из файла и вывести в другой файл в столбик все слова, которые начинаются с буквы А.
12. Прочитать текст из файла, заменить везде слово «паровоз» на слово «пароход» и записать в другой файл.
13. В файле записаны данные о результатах сдачи экзамена. Каждая строка содержит фамилию, имя и количество баллов, разделенные пробелами:
<Фамилия> <Имя> <Количество баллов>
Вывести фамилии и имена тех учеников, которые получили больше 80 баллов.
14. В предыдущей задаче добавить к списку нумерацию, например:
1) Иванов Вася
2) Петров Петя
15. В той же задаче сократить имя до одной буквы и поставить перед фамилией:
1) В. Иванов
2) П. Петров
16. В той же задаче отсортировать список по алфавиту (по фамилии).
17. *В той же задаче отсортировать список по убыванию полученного балла (вывести балл в выходной файл).