МОУ Павловская СОШ № 3.

РАССМОТРЕНО

На заседании методического совета

Протокол №\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

УТВЕРЖДЕНО

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Экзамен по информатике и ИКТ**

**в 10 «Б» классе**

**Учитель: Пинчукова М.В.**

**2012Введение**

Часть 1 (1- 13 вопросы) включает 14 заданий с выбором ответа. К каждому заданию дается четыре ответа, из которых только один правильный.

 Часть 2 (14-22 вопросы) состоит из 7-х заданий с кратким ответом (к этим заданиям вы должны самостоятельно сформулировать и записать ответ).

  Часть 3 (23-24 вопросы) состоит из 2-х заданий. Для выполнения заданий этой части вам необходимо написать развернутый ответ в произвольной форме.

За каждый правильный ответ в зависимости от сложности задания дается один. За задания из части 3 дается 3 балла.

**Критерии оценивания работ**

21– 28 баллов - оценка 5

11 – 20 баллов – оценка 4

6 – 10 баллов – оценка 3

0 – 5 баллов – оценка 2

В экзаменационных заданиях используются следующие соглашения:

 1.     Обозначения для логических связок (операций):

a) отрицание (инверсия, логическое НЕ) обозначается 
(например, А);

b) конъюнкция (логическое умножение, логическое И) обозначается /\
(например, А /\ В) либо & (например, А & В);

c) дизъюнкция (логическое сложение, логическое ИЛИ) обозначается \/
(например, А \/ В) либо | (например, А | В);

d) следование (импликация) обозначается –> (например, А –> В);

e) символ 1 используется для обозначения истины (истинного высказывания); символ 0 – для обозначения лжи (ложного высказывания).

 2. Два логических выражения, содержащих переменные, называются равносильными (эквивалентными), если значения этих выражений совпадают при любых значениях переменных. Так, выражения А –> В и (А) \/ В равносильны, а А \/ В и А /\ В – нет (значения выражений разные, например, при А = 1, В = 0).

 3. Приоритеты логических операций: инверсия (отрицание), конъюнкция (логическое умножение), дизъюнкция (логическое сложение), импликация (следование), эквивалентность (равносильность). Таким образом, А /\ В \/ С /\ D совпадает с ((А) /\ В) \/ (С /\ D). Возможна запись А /\ В /\ С вместо (А /\ В) /\ С. То же относится и к дизъюнкции: возможна запись А \/ В \/ С вместо (А \/ В) \/ С.

**Примеры экзаменационных заданий**

# А1 (базовый уровень, время – 1 мин)

**Тема**: Системы счисления и двоичное представление информации в памяти компьютера.

**Что нужно знать**:

* перевод чисел между десятичной, двоичной, восьмеричной и шестнадцатеричной системами счисления (см. презентацию «Системы счисления»)

|  |
| --- |
| **Полезно помнить, что в двоичной системе:*** четные числа оканчиваются на 0, нечетные – на 1;
* числа, которые делятся на 4, оканчиваются на 00, и т.д.; числа, которые делятся на 2k, оканчиваются на *k* нулей
* если число N принадлежит интервалу 2k-1 ≤ N < 2k, в его двоичной записи будет всего *k* цифр, например, для числа **125**:

 26 = 64 ≤ **125** < 128 = 27, 125 = 11111012 (7 цифр)* числа вида 2k записываются в двоичной системе как единица и *k* нулей, например:

 16 = 24 = 100002* числа вида 2k-1записываются в двоичной системе *k* единиц, например:

 15 = 24-1 = 11112* если известна двоичная запись числа N, то двоичную запись числа 2·N можно легко получить, приписав в конец ноль, например: 15 = 11112, 30 = 111102, 60 = 1111002, 120 = 11110002
 |

* отрицательные целые числа хранятся в памяти в двоичном дополнительном коде (подробнее см. презентацию «Компьютер изнутри»)
* для перевода отрицательного числа **(-a)** в двоичный дополнительный код нужно сделать следующие операции:
	+ перевести число **a-1** в двоичную систему счисления
	+ сделать инверсию битов: заменить все нули на единицы и единицы на нули в пределах разрядной сетки (см. пример далее)

### Пример задания:

*Сколько единиц в двоичной записи числа 1025?*

1) 1 2) 2 3) 10 4) 11

# А2 (базовый уровень, время – 1 мин)

**Тема**: Системы счисления и двоичное представление информации в памяти компьютера.

# А3 (базовый уровень, время – 1 мин)

**Тема**: Системы счисления и двоичное представление информации в памяти компьютера.

# A4 (базовый уровень, время – 1 мин)

**Тема**: Файловая система.

**Что нужно знать**:

* данные на дисках хранятся в виде файлов (наборов данных, имеющих имя)
* чтобы было удобнее разбираться с множеством файлов, их объединяют в каталоги (в *Windows* каталоги называются «папками»)
* каталог можно воспринимать как контейнер, в котором размещаются файлы и другие каталоги, которые называются подкаталогами или *вложенными* каталогами (они находятся внутри другого каталога, вложены в него)
* каталоги организованы в многоуровневую (иерархическую) структуру, которая называется «деревом каталогов»
* главный каталог диска (который пользователь видит, «открыв» диск, например, в Проводнике *Windows* или аналогичной программе) называется *корневым* каталогом или «корнем» диска, он обозначается буквой логического диска, за которой следует двоеточие и знак «\» (обратный слэш[[1]](#footnote-2)); например, **A:\** – это обозначение корневого каталога диска А
* каждый каталог (кроме корневого) имеет (один единственный!) «родительский» каталог – этот тот каталог, внутри которого находится данный каталог
* полный адрес каталога – это перечисление каталогов, в которые нужно войти, чтобы попасть в этот каталог (начиная с корневого каталога диска); например
 **С:\USER\BIN\SCHOOL**
* полный адрес файла состоит из адреса каталога, в котором он находится, символа «\» и имени файла, например
 **С:\USER\BIN\SCHOOL\Вася.txt**
* маска служит для обозначения (выделения) группы файлов, имена которых имеют общие свойства, например, общее расширение
* в масках, кроме «обычных» символов (допустимых в именах файлов) используются два специальных символа: звездочка «\*» и знак вопроса «?»;
* звездочка «\*» обозначает любой количество любых символов, в том числе, может обозначать пустую последовательность;
* знак вопроса «?» обозначает ровно один любой символ
* при выводе списка имен файлов они могут быть отсортированы по имени, типу (расширению), дате последнего изменения, размеру; это не меняет их размещения на диске;
* если установлена сортировка по имени или типу, сравнение идет по кодам символов, входящих в имя или в расширение

### Пример задания:

*Определите, какое из указанных имен файлов удовлетворяет маске:* **?hel\*lo.c?\***

1) **hello.c** 2) **hello.cpp** 3) **hhelolo.cpp** 4) **hhelolo.c**

# A5 (базовый уровень, время – 1 мин)

**Тема**: Файловая система.

# A6 (повышенный уровень, время – 2 мин)

**Тема**: Основные понятия математической логики.

**Про обозначения**

К сожалению, обозначения логических операций И, ИЛИ и НЕ, принятые в «серьезной» математической логике (**∧**,**∨**,**¬**), неудобны, интуитивно непонятны и никак не проявляют аналогии с обычной алгеброй. Автор, к своему стыду, до сих пор иногда путает **∧** и **∨**. Поэтому на его уроках операция «НЕ» обозначается чертой сверху, «И» – знаком умножения (поскольку это все же логическое умножение), а «ИЛИ» – знаком «+» (логическое сложение).
В разных учебниках используют разные обозначения. К счастью, в начале задания ЕГЭ приводится расшифровка закорючек (**∧**, **∨**,**¬**), что еще раз подчеркивает проблему. Далее во всех решениях приводятся два варианта записи.

**Что нужно знать**:

* условные обозначения логических операций

**¬ A, **  не A (отрицание, инверсия)

**A ∧ B, ** A и B (логическое умножение, конъюнкция)

**A ∨ B, **  A или B (логическое сложение, дизъюнкция)

**A** → **B**  импликация (следование)

* таблицы истинности логических операций «И», «ИЛИ», «НЕ», «импликация» (см. презентацию «Логика»)
* операцию «импликация» можно выразить через «ИЛИ» и «НЕ»:

**A** → **B = ¬ A ∨ B** или в других обозначениях **A** → **B = **

* если в выражении нет скобок, сначала выполняются все операции «НЕ», затем – «И», затем – «ИЛИ», и самая последняя – «импликация»
* иногда полезны формулы де Моргана[[2]](#footnote-3):

**¬ (A ∧ B) = ¬ A ∨ ¬ B **

**¬ (A ∨ B) = ¬ A ∧ ¬ B **

### Пример задания:

*Какое из приведённых имен удовлетворяет логическому условию:*(первая буква согласная → вторая буква согласная) /\ (предпоследняя буква гласная → последняя буква гласная)*?*

1) КРИСТИНА 2) МАКСИМ 3) СТЕПАН 4) МАРИЯ

# A7 (повышенный уровень, время – 2 мин)

**Тема**: Основные понятия математической логики.

# А8 (повышенный уровень, время – 5 мин)

**Тема**: Работа с массивами и матрицами в языке программирования

**Что нужно знать**:

* работу цикла **for** (цикла с переменной)
* массив – это набор однотипных элементов, имеющих общее имя и расположенных в памяти рядом
* для обращения к элементу массива используют квадратные скобки, запись **A[i]** обозначает элемент массива **A** с номером (индексом) **i**
* матрица (двухмерный массив) – это прямоугольная таблица однотипных элементов
* если матрица имеет имя A, то обращение **A[i,k]** обозначает элемент, расположенный на пересечении строки **i** и столбца **k**
* элементы, у которых номера строки и столбца совпадают, расположены на главной диагонали

|  |  |  |  |
| --- | --- | --- | --- |
| A[1,1] |  |  |  |
|  | A[2,2] |  |  |
|  |  | A[3,3] |  |
|  |  |  | A[4,4] |

* выше главной диагонали расположены элементы, у которых номер строки **меньше** номера столбца:

|  |  |  |  |
| --- | --- | --- | --- |
|  | A[1,2] | A[1,3] | A[1,4] |
|  |  | A[2,3] | A[2,4] |
|  |  |  | A[3,4] |
|  |  |  |  |

* ниже главной диагонали расположены элементы, у которых номер строки **больше** номера столбца:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| A[2,1] |  |  |  |
| A[3,1] | A[3,2] |  |  |
| A[4,1] | A[4,2] | A[4,3] |  |

### Ещё пример задания:

*В программе используется одномерный целочисленный массив A с индексами от 0 до 9. Ниже представлен фрагмент программы, записанный на разных языках программирования, в котором значения элементов сначала задаются, а затем меняются.*

**for i:=0 to 9 do**

 **A[i]:=9-i;**

**for i:=0 to 4 do begin**

 **k:=A[i];**

 **A[i]:=A[9-i];**

 **A[9-i]:=k;**

**end;**

*Чему будут равны элементы этого массива после выполнения фрагмента программы?*

 1) 9 8 7 6 5 4 3 2 1 0

 2) 0 1 2 3 4 5 6 7 8 9

 3) 9 8 7 6 5 5 6 7 8 9

 4) 0 1 2 3 4 4 3 2 1 0

# А9 (повышенный уровень, время – 5 мин)

**Тема**: Работа с массивами и матрицами в языке программирования

# A10 (базовый уровень, время – 2 мин)

**Тема**: Использование информационных моделей (таблицы, диаграммы, графики).
 Перебор вариантов, выбор лучшего по какому-то признаку.

**Что нужно знать**:

* в принципе, особых дополнительных знаний, кроме здравого смысла и умения перебирать варианты (не пропустив ни одного!) здесь, как правило, не требуется
* полезно знать, что такое *граф* (это набор вершин и соединяющих их ребер) и как он описывается в виде таблицы, хотя, как правило, все необходимые объяснения даны в формулировке задания
* чаще всего используется *взвешенный граф*, где с каждым ребром связано некоторое число (вес), оно может обозначать, например, расстояние между городами или стоимость перевозки
* рассмотрим граф (рисунок слева), в котором 5 вершин (A, B, C, D и E); он описывается таблицей, расположенной в центре; в ней, например, число 4 на пересечении строки В и столбца С означает, что, во-первых, есть ребро, соединяющее В и С, и во-вторых, вес этого ребра равен 4; пустая клетка на пересечении строки А и столбца В означает, что ребра из А в В нет

1

2

4

2

3

1

2

4

2

3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | Е |
| A |  |  | 3 | 1 |  |
| B |  |  | 4 |  | 2 |
| C | 3 | 4 |  |  | 2 |
| D | 1 |  |  |  |  |
| Е |  | 2 | 2 |  |  |

* обратите внимание, что граф по заданной таблице (она еще называется *весовой матрицей*) может быть нарисован по-разному; например, той же таблице соответствует граф, показанный на рисунке справа от нее
* в приведенном примере матрица симметрична относительно главной диагонали; это может означать, например, что стоимости перевозки из В в С и обратно равны (это не всегда так)
* желательно научиться быстро (и правильно) строить граф по весовой матрице и наоборот

### Пример задания:

*Между населёнными пунктами A, B, C, D, E, F построены дороги, протяжённость которых приведена в таблице. (Отсутствие числа в таблице означает, что прямой дороги между пунктами нет.)*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E | F |
| A |  | 2 | 4 |  |  |  |
| B | 2 |  | 1 |  | 7 |  |
| C | 4 | 1 |  | 3 | 4 |  |
| D |  |  | 3 |  | 3 |  |
| E |  | 7 | 4 | 3 |  | 2 |
| F |  |  |  |  | 2 |  |

*Определите длину кратчайшего пути между пунктами A и F (при условии, что передвигаться можно только по построенным дорогам).*

1) 9 2) 10 3) 11 4) 12

# A11 (базовый уровень, время – 2 мин)

**Тема**: Использование информационных моделей (таблицы, диаграммы, графики).
 Перебор вариантов, выбор лучшего по какому-то признаку.

# A12 (базовый уровень, время – 2 мин)

**Тема**: Проверка закономерностей методом рассуждений.

**Что нужно знать**:

* в общем-то, никаких знаний из курса информатики здесь не требуется, эту задачу можно давать детям начальной школы для развития логического мышления

### Пример задания:

*Автомат получает на вход два трехзначных числа. По этим числам строится новое число по следующим правилам. Вычисляются три числа – сумма старших разрядов заданных трехзначных чисел, сумма средних разрядов этих чисел, сумма младших разрядов.*

*Полученные три числа записываются друг за другом в порядке убывания (без разделителей).*

***Пример.***

*Исходные трехзначные числа: 835, 196. Поразрядные суммы: 9, 12, 11. Результат: 12119*

*Определите, какое из следующих чисел может быть результатом работы автомата.*

1) 151303 2) 161410 3) 191615 4) 121613

# A13 (базовый уровень, время – 2 мин)

**Тема**: Проверка закономерностей методом рассуждений.

# B1 (базовый уровень, время – 2 мин)

**Тема**: Анализ последовательностей, системы счисления.

**Что нужно знать**:

* русский алфавит
* принципы работы с числами, записанными в позиционных системах счисления

### Пример задания:

*Все 5-буквенные слова, составленные из букв А, О, У, записаны в алфавитном порядке.*

*Вот начало списка:*

**1. ААААА**

**2. ААААО**

**3. ААААУ**

**4. АААОА**

**……**

*Запишите слово, которое стоит на 240-м месте от начала списка.*

# B2 (базовый уровень, время – 2 мин)

**Тема**: Оператор присваивания в языке программирования

**Что нужно знать**:

* переменная – это величина, которая имеет имя, тип и значение; переменная может изменяться во время выполнения программы
* оператор присваивания служит для записи значения в переменную
* если в переменную записывают новое значение, старое стирается
* знаки **+**, **-**, **\***, **/** используются для обозначения операций сложения, вычитания, умножения и деления
* запись вида **a div b** означает результат целочисленного деления **a** на **b** (остаток отбрасывается)
* запись вида **a mod b** означает остаток от деления **a** на **b**
* запись вида **a := b + 2\*c + 3;** означает «вычислить значения выражения справа от знака присваивания := и записать результат в переменную **a**»; при этом значения других переменных (кроме **a**) не изменяются
* для многократного выполнения одинаковых операций используют циклы;
* цикл с переменной выполняется N раз, в этом примере переменная **i** принимает последовательно все значения от 1 до N с шагом 1

**for i:=1 to N do begin**

 **{ что-то делаем }**

**end;**

* цикл с условием выполняется до тех пор, пока условие в заголовке цикла не нарушится;

**while { условие } do begin**

 **{ что-то делаем }**

**end;**

* главная опасность при использовании цикла с условием – **зацикливание**; эта такая ситуация, когда цикл работает бесконечно долго из-за того, что условие все время остается истинным

### Пример задания:

*Определите значение переменной c после выполнения следующего фрагмента программы.*

**a := 5;**

**a := a + 6;**

**b := –a;**

**c := a – 2\*b;**

# B3 (базовый уровень, время – 2 мин)

**Тема**: Оператор присваивания в языке программирования

# B4 (повышенный уровень, время – 2 мин)

**Тема**: Кодирование чисел. Системы счисления.

**Что нужно знать**:

* принципы кодирования чисел в позиционных системах счисления
* чтобы перевести число, скажем, 12345N, из системы счисления с основанием  в десятичную систему, нужно умножить значение каждой цифры на  в степени, равной ее разряду:

**N0 = 1**

4 3 2 1 0 ← разряды

**1 2 3 4 5N = 1·N4 + 2·N3 + 3·N2 + 4·N1 + 5·N0**

* последняя цифра записи числа в системе счисления с основанием  – это остаток от деления этого числа на 
* две последние цифры – это остаток от деления на , и т.д.

### Пример задания:

*Запись числа 6710 в системе счисления с основанием N оканчивается на 1 и содержит 4 цифры. Укажите основание этой системы счисления N.*

# B5 (повышенный уровень, время – 3 мин)

**Тема**: Графы. Поиск путей

**Что нужно знать**:

* если в город R можно приехать только из городов X, Y, и Z, то число различных путей из города A в город R равно сумме числа различных путей проезда из A в X, из A в Y и из A в Z, то есть

,

где  обозначает число путей из вершины A в некоторую вершину Q

* число путей конечно, если в графе нет циклов – замкнутых путей

### Пример задания:

 *На рисунке – схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, И, К. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город К?*

Г

В

А

К

Е

Б

Д

Ж

И

# B6 (повышенный уровень, время – 3 мин)

**Тема**: Графы. Поиск путей

# B7 (повышенный уровень, время – 2 мин)

**Тема**: Составление запросов для поисковых систем с использованием логических выражений.

**Что нужно знать**:

* таблицы истинности логических операций «И», «ИЛИ», «НЕ» (см. презентацию «Логика»)
* если в выражении нет скобок, сначала выполняются все операции «НЕ», затем – «И», затем – «ИЛИ»
* логическое произведение A∙B∙C∙… равно 1 (выражение истинно) только тогда, когда все сомножители равны 1 (а в остальных случаях равно 0)
* логическая сумма A+B+C+… равна 0 (выражение ложно) только тогда, когда все слагаемые равны 0 (а в остальных случаях равна 1)
* правила преобразования логических выражений (законы алгебры логики):

|  |  |  |
| --- | --- | --- |
| Закон | Для **И** | Для **ИЛИ** |
| двойного отрицания |  |
| исключения третьего |  |  |
| исключения констант | A · 1 = A; A · 0 = 0 | A + 0 = A; A + 1 = 1 |
| повторения | A · A = A | A + A = A |
| поглощения | A · (A + B) = A | A + A · B = A |
| переместительный | A · B = B · A | A + B = B + A |
| сочетательный | A · (B · C) = (A · B) · C | A + (B + C) = (A + B) + C |
| распределительный | A + B · C = (A + B) · (A + C) | A · (B + C) = A · B + A · C |
| де Моргана |  |  |

* ввод какого-то слова (скажем, **кергуду**) в запросе поисковой системы означает, что пользователь ищет Web-страницы, на которых встречается это слово
* операция «И» всегда **ограничивает** поиск, то есть, в ответ на запрос **кергуду И бамбарбия** поисковый сервер выдаст **меньше** страниц, чем на запрос **кергуду**, потому что будет искать страницы, на которых есть оба этих слова одновременно
* операция «ИЛИ» всегда **расширяет** поиск, то есть, в ответ на запрос
**кергуду ИЛИ бамбарбия** поисковый сервер выдаст **больше** страниц, чем на запрос **кергуду**, потому что будет искать страницы, на которых есть хотя бы одно из этих слов (или оба одновременно)
* если в запросе вводится фраза в кавычках, поисковый сервер ищет страницы, на которых есть в точности эта фраза, а не просто отдельные слова; взятие словосочетания в кавычки **ограничивает** поиск, то есть, в ответ на запрос **"кергуду бамбарбия"** поисковый сервер выдаст **меньше** страниц, чем на запрос **кергуду бамбарбия**, потому что будет искать только те страницы, на которых эти слова стоят одно за другим

### Пример задания:

*В таблице приведены запросы к поисковому серверу. Расположите номера запросов в порядке возрастания количества страниц, которые найдет поисковый сервер по каждому запросу. Для обозначения логической операции «ИЛИ» в запросе используется символ* |*, а для логической операции «И» – &.*

1) **принтеры & сканеры & продажа**

2) **принтеры & сканеры**

3) **принтеры | сканеры**

4) **принтеры | сканеры | продажа**

# B8 (базовый уровень, время – 1 мин)

**Тема**: Кодирование текстовой информации. Кодировка ASCII. Основные кодировки кириллицы.

**Что нужно знать**:

* все символы кодируются одинаковым числом бит[[3]](#footnote-4) (алфавитный подход)
* чаще всего используют кодировки, в которых на символ отводится 8 бит (8-битные) или 16 бит (16-битные)
* при измерении количества информации принимается, что в одном байте 8 бит, а в одном килобайте (1 Кбайт) – 1024 байта, в мегабайте (1 Мбайт) – 1024 Кбайта
* после знака препинания внутри (не в конце!) текста ставится пробел
* чтобы найти информационный объем текста *I*, нужно умножить количество символов *N* на число бит на символ *K*: 
* две строчки текста не могут занимать 100 Кбайт в памяти
* при кодировании текста каждому символу ставится в соответствие свой код
* символы-цифры в кодовой таблице идут подряд в порядке возрастания, от «0» до «9» (символ «0» имеет код 48, символ «9» – код 57)
* большие (прописные) латинские буквы в кодовой таблице идут подряд в алфавитном порядке от «A» до «Z»
* маленькие (строчные) латинские буквы в таблице кодировки идут подряд в алфавитном порядке, от «a» до «z»
* в большинстве кодовых таблиц (но не во всех!) русские буквы, как заглавные, так и строчные, также расположены по алфавиту (за исключением буквы «Ё»)

### Ещё пример задания:

*Автоматическое устройство осуществило перекодировку информационного сообщения на русском языке длиной в 20 символов, первоначально записанного в 2-байтном коде Unicode, в 8-битную кодировку КОИ-8. На сколько бит уменьшилась длина сообщения? В ответе запишите только число.*

# B9 (базовый уровень, время – 1 мин)

**Тема**: Кодирование текстовой информации. Кодировка ASCII. Основные кодировки кириллицы.

# C1 (высокий уровень, время – 30 мин)

**Тема**: Обработка массива (написать программу из 10-15 строк на языке программирования или алгоритм на естественном языке).

**Что нужно знать**:

* *массив* – это набор однотипных элементов, имеющих общее имя и расположенных в памяти рядом
* для обращения к элементу массива используют квадратные скобки, запись **A[i]** обозначает элемент массива **A** с номером (индексом) **i**
* для обработки всех элементов массива используется цикл вида[[4]](#footnote-5)

**for i:=1 to N do begin**

 **{ что-то делаем с элементом A[i] }**

**end;**

переменная **i** обозначает номер текущего элемента массива, она меняется от 1 до N с шагом 1, то есть мы «проходим» последовательно все элементы

* *матрица* (двухмерный массив) – это прямоугольная таблица однотипных элементов
* если матрица имеет имя **A**, то обращение **A[i,k]** обозначает элемент, расположенный на пересечении строки **i** и столбца **k**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **k** |  |
|  |  |  |  |
|  |  |  |  |
| **i** |  | **A[i,k]** |  |
|  |  |  |  |

* каждая строка матрицы – это обычный (одномерный, линейный) массив; для того, чтобы обработать строку **i** в матрице из **M** столбцов, нужно использовать цикл, в котором меняется номер столбца **k**:

**for k:=1 to M do begin**

 **{ что-то делаем с элементом A[i,k] }**

**end;**

* каждый столбец матрицы – это обычный (одномерный, линейный) массив; для того, чтобы обработать столбец **k** в матрице из **N** строк, нужно использовать цикл, в котором изменяется номер строки **i**:

**for i:=1 to N do begin**

 **{ что-то делаем с элементом A[i,k] }**

**end;**

### Пример задания:

*Дан целочисленный массив из 20 элементов. Элементы массива могут принимать целые значения от 0 до 1000. Опишите на русском языке или на одном из языков программирования алгоритм, позволяющий найти и вывести минимальное значение среди элементов массива, которые имеют чётное значение и не делятся на три. Гарантируется, что в исходном массиве есть хотя бы один элемент, значение которого чётно и не кратно трем.*

*Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но использовать все описанные переменные не обязательно.*

***Естественный язык:***

**Объявляем массив A из 20 элементов.**

**Объявляем целочисленные переменные I, J, MIN.**

**В цикле от 1 до 20 вводим элементы массива A с 1-го по 20-й.**

***Паскаль:***

**const N=20;**

**var a: array [1..N] of integer;**

 **i, j, min: integer;**

**begin**

 **for i:=1 to N do**

 **readln(a[i]);**

 **…**

**end.**

# C1 (высокий уровень, время – 30 мин)

**Тема**: Обработка массива (написать программу из 10-15 строк на языке программирования или алгоритм на естественном языке).

**Рекомендуемая литература**

1. Угринович Н.Д. Информатика и ИКТ: Учебник для 10 класса. – М.: БИНОМ. Лаборатория знаний, 2007-2009.
2. Угринович Н.Д. Босова Л.Л., Михайлова Н.И. Практикум по информатике и информационным технологиям. Учебное пособие для общеобразовательных учреждений. – М.: БИНОМ Лаборатория знаний, 2006.
3. Угринович Н.Д. Преподавание курса «Информатика и ИКТ» в основной и старшей школе (7-11 кл.).- М.: БИНОМ Лаборатория знаний, 2006.

Дополнительная литература:

1. Белоусова Л.И. Сборник задач по курсу информатики. – М.: Издательство «Экзамен», 2008.
2. Босова Л.Л. и др. Обработка текстовой информации: Дидактические материалы.- М.: БИНОМ Лаборатория знаний, 2007.
3. Богомолова О.Б. Практические работы по MS Excel на уроках информатики. – М.: БИНОМ Лаборатория знаний, 2007.
4. Информатика. Задачник-практикум в 2 т./Под ред. Г. Семакина, Е.К. Хеннера. - М.: БИНОМ Лаборатория знаний, 2007
5. Сафронов И.К. Задачник-практикум по информатике. – СПб: БХВ-Петербург, 2002.
1. [↑](#footnote-ref-2)
2. [↑](#footnote-ref-3)
3. [↑](#footnote-ref-4)
4. [↑](#footnote-ref-5)