**Теоретические основы информатики**

**Системы счисления**

Непозиционные и позиционные системы счисления. Алфавит. Основание. Развёрнутая форма записи числа. Перевод в десятичную систему чисел, записанных в других системах счисления.

Римская система счисления.

Двоичная система счисления. Перевод целых чисел в пределах от 0 до 1024 в двоичную систему счисления. Восьмеричная система счисления. Перевод чисел из восьмеричной системы в двоичную и десятичную системы и обратно. Шестнадцатеричная система счисления. Перевод чисел из шестнадцатеричной системы в двоичную, восьмеричную и десятичную системы и обратно.

Арифметические операции в двоичной системе счисления.

**Элементы математической логики**

Логические высказывания. Логические значения высказываний. Элементарные и составные высказывания. Логические операции: «и» (конъюнкция, логическое умножение), «или» (дизъюнкция, логическое сложение), «не» (логическое отрицание). Приоритет логических операций. Определение истинности составного высказывания, если известны значения истинности входящих в него элементарных высказываний. Логические выражения. Правила записи логических выражений. Построение таблиц истинности логических выражений.

Логические элементы. Знакомство с логическими основами компьютера.

**Исполнители и алгоритмы. Алгоритмические конструкции**

Понятие алгоритма. Исполнители алгоритмов. Алгоритм как план управления исполнителем.

Свойства алгоритма. Способы записи алгоритма (словесный, в виде блок-схемы, программа).

Алгоритмические конструкции. Конструкция «следование». Линейный алгоритм. Ограниченность линейных алгоритмов: невозможность предусмотреть зависимость последовательности выполняемых действий от исходных данных.

Конструкция «ветвление»: полная и неполная формы. Выполнение и невыполнение условия (истинность и ложность высказывания). Простые и составные условия.

Конструкция «повторения»: циклы с заданным числом повторений, с условием выполнения, с переменной цикла.

Разработка для формального исполнителя алгоритма, приводящего к требуемому результату при конкретных исходных данных. Разработка несложных алгоритмов с использованием циклов и ветвлений для управления формальными исполнителями, такими как Робот, Черепашка. Выполнение алгоритмов вручную и на компьютере. Синтаксические и логические ошибки. Отказы.

**Типовая работа по информатике**

**за курс 8 класса**

**МБОУ СШ №1**

**г.Вилючинск**

1. Переведите в десятичную систему двоичное число 1000011002.
2. Переведите в двоичную систему десятичное число 13710.
3. Переведите в десятичную систему следующие числа:

а) 1518; б) 2С16.

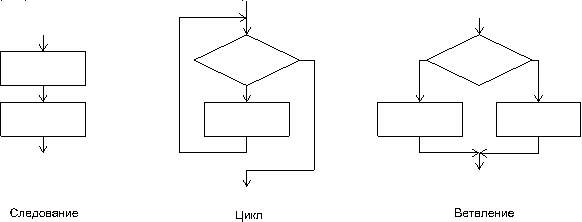
1. Укажите число, записанное неверно

а) 101102; б) 2123; в) 4656; г) 6058.

1. Система счисления называется… если количественный эквивалент цифры в числе не зависит от её положения в записи числа.

а) позиционной; б) непозиционной.

1. Операция **дизъюнкция** называется иначе:  
   а) логическое умножение б) логическое сложение   
   в) логическое следование г) логическое отрицание
2. Для какого имени верно высказывание   
   **НЕ** (Первая буква согласная) **И** (Последняя буква гласная)?
3. ИВАН
4. ФЁДОР
5. ПОЛИНА
6. АННА
7. Алгоритм – это:
8. правила выполнения определенных действий;
9. описание последовательности действий, строгое исполнение которых приводит к решению поставленной задачи за конечное число шагов;
10. ориентированный граф, указывающий порядок выполнения некоторого набора команд;
11. набор команд для компьютера.
12. К какому виду алгоритмов можно отнести алгоритм, блок-схема которого указана ниже?



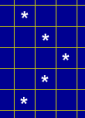
1. линейный
2. разветвляющийся с неполным ветвлением
3. разветвляющийся с полным ветвлением
4. циклический
5. Вычислите: ((1 & 0) v 1) & (1 v A).
6. Составьте таблицу истинности для следующей логической функции:

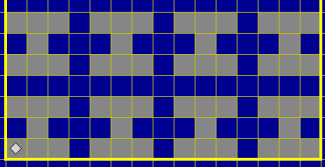
F = X & ¬Y v ¬X & Y.

1. В таб­ли­це приведены за­про­сы к по­ис­ко­во­му серверу. Для каж­до­го запроса ука­зан его код — со­от­вет­ству­ю­щая буква от А до Г. Рас­по­ло­жи­те коды за­про­сов слева на­пра­во в по­ряд­ке возрастания ко­ли­че­ства страниц, ко­то­рые нашёл по­ис­ко­вый сервер по каж­до­му запросу. По всем за­про­сам было най­де­но разное ко­ли­че­ство страниц. Для обо­зна­че­ния логической опе­ра­ции «ИЛИ» в за­про­се используется сим­вол «|», а для ло­ги­че­ской операции «И» — «&»:

|  |  |
| --- | --- |
| **Код** | **Запрос** |
| А | Рыжий | Чест­ный | Влюблённый |
| Б | (Рыжий & Честный)|Влюблённый |
| В | Рыжий & Честный |
| Г | Рыжий & Чест­ный & Влюблённый |

1. Составьте алгоритм закрашивания клеток поля, отмеченных звездочкой. Начальное положение Робота где-то в центре поля.



1. Составьте алгоритм рисования узора. Начальное положение Робота отмечено символом ◊.
2. Исполнитель Робот умеет перемещаться по лабиринту, начерченному на плоскости, разбитой на клетки. Между соседними (по сторонам) клетками может стоять стена, через которую Робот пройти не может. У Робота есть девять команд. Четыре команды  — это команды-приказы:

**вверх    вниз    влево    вправо**

При выполнении любой из этих команд Робот перемещается на одну клетку соответственно: вверх ↑, вниз ↓, влево ←, вправо →. Если Робот получит команду передвижения сквозь стену, то он разрушится. Также у Робота есть команда **закрасить**,при которойзакрашивается клетка, в которой Робот находится в настоящий момент.

Еще четыре команды  — это команды проверки условий. Эти команды проверяют, свободен ли путь для Робота в каждом из четырех возможных направлений:

**сверху свободно   снизу свободно   слева свободно   справа свободно**

Эти команды можно использовать вместе с условием «**eсли**», имеющим следующий вид:

**если** *условие* **то**

*последовательность команд*

**все**

Здесь *условие*  — одна из команд проверки условия. *Последовательность* *команд*  — это одна или несколько любых команд-приказов.

Например, для передвижения на одну клетку вправо, если справа нет стенки и закрашивания клетки, можно использовать такой алгоритм:

**если справа свободно то**

**вправо**

**закрасить**

**все**

В одном условии можно использовать несколько команд проверки условий, применяя логические связки **и**, **или**, **не**, например:

**если (справа свободно) и (не снизу свободно) то**

**вправо**

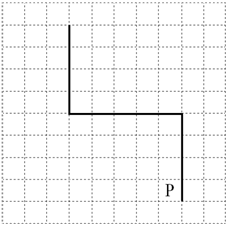
**все**

Для повторения последовательности команд можно использовать цикл «**пока**», имеющий следующий вид:

**нц пока**  *условие*

*последовательность команд*

**кц**

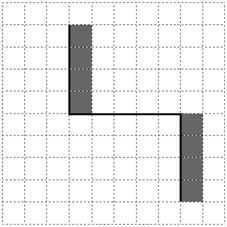
Например, для движения вправо, пока это возможно, можно использовать следующий алгоритм:

**нц пока справа свободно**

**вправо**

**кц**

**Выполните задание.**

На бесконечном поле имеются две вертикальные стены и одна горизонтальная, соединяющая нижний конец левой и верхний конец правой вертикальных стен. **Длины стен неизвестны**. Робот находится в клетке, расположенной слева от нижнего края правой вертикальной стены, рядом со стеной. На рисунке указан один из возможных способов расположения стен и Робота (Робот обозначен буквой «Р»).

Напишите для Робота алгоритм, закрашивающий все клетки, примыкающие к вертикальным стенам справа. Робот должен закрасить только клетки, удовлетворяющие данному условию. Например, для приведенного выше рисунка Робот должен закрасить следующие клетки (см. рис.).

Конечное расположение Робота может быть произвольным. Алгоритм должен решать задачу для произвольного размера поля и любого допустимого расположения стен внутри прямоугольного поля. При исполнении алгоритма Робот не должен разрушиться, выполнение алгоритма должно завершиться. Алгоритм может быть выполнен в среде формального исполнителя или  записан в текстовом редакторе. Сохраните алгоритм в формате программы Кумир или в текстовом файле. Название файла и каталог для сохранения Вам сообщат организаторы экзамена.