

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ВГУ»)

УТВЕРЖДАЮ



Председатель приемной комиссии

Ректор

Д.А. Ендовицкий

29 октября 2021

**ПРОГРАММА ВСТУПИТЕЛЬНОГО ИСПЫТАНИЯ ПРИ ПРИЕМЕ
НА ОБУЧЕНИЕ ПО ПРОГРАММАМ БАКАЛАВРИАТА И СПЕЦИАЛИТЕТА**

ИНФОРМАТИКА, АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ

Воронеж

2021

Программа разработана на основе ФГОС среднего профессионального образования. В первом разделе программы перечислены основные понятия, которыми должен владеть поступающий.

Во втором разделе указано, какие навыки и умения требуются от поступающего.

Для решения экзаменационных задач достаточно уверенного владения теми понятиями и свойствами, которые перечислены в настоящей программе.

1. Основные понятия и факты Информатика

1. Тексты и кодирование (Равномерные и неравномерные коды. Префиксные коды. Условие Фано. Обратное условие Фано. Алгоритмы декодирования при использовании префиксных кодов. Примеры кодов, используемых для кодирования текстов: ASCII, стандарт UNICODE. Кодирование кириллицы и других национальных алфавитов; кодовые страницы. Пропускная способность и помехозащищенность канала связи. Кодирование сообщений в современных средствах передачи данных. Искажение информации при передаче по каналам связи. Коды с возможностью обнаружения и исправления ошибок).
2. Дискретизация (Измерения и дискретизация. Частота и разрядность измерений. Универсальность дискретного представления информации. Дискретное представление звуковых данных. Многоканальная запись. Размер файла, полученного в результате записи звука. Дискретное представление графической информации. Цветовые модели. Модели RGB и CMYK. Модели HSB и CMY. Глубина кодирования. Сжатие данных при хранении графической и звуковой информации).
3. Системы счисления (Свойства позиционной записи числа: количество цифр в записи, признак делимости числа на основание системы счисления. Алгоритм перевода десятичной записи числа в запись в позиционной системе с заданным основанием. Алгоритмы построения записи числа в позиционной системе счисления с заданным основанием и вычисления числа по строке, содержащей запись этого числа в позиционной системе счисления с заданным основанием. Арифметические действия в позиционных системах счисления. Краткая и развернутая форма записи смешанных чисел в позиционных системах счисления. Перевод смешанного числа в позиционную систему счисления с заданным основанием. Представление целых и вещественных чисел в памяти компьютера. Компьютерная арифметика).
4. Дискретные объекты (Ориентированные и неориентированные графы; цикл и ациклические графы; начальная вершина (источник) и конечная вершина (сток) в ориентированном ациклическом графе; расстояние между вершинами. Алгоритмические задачи, связанные с анализом графов (примеры: задача построения оптимального пути между вершинами ориентированного ациклического графа; задача определения количества различных путей между вершинами). Деревья. Поддерево; обход узлов дерева в глубину. Упорядоченные деревья (деревья, в которых упорядочены ребра, выходящие из одного узла). Использование деревьев при решении алгоритмических задач (примеры: анализ работы рекурсивных алгоритмов, разбор арифметических и логических выражений). Использование графов, деревьев, списков при описании объектов и процессов окружающего мира).

Алгоритмизация

1. Алгоритмы и структуры данных (Алгоритмы исследования квадратного уравнения с целыми и вещественными коэффициентами. Алгоритмы анализа и преобразования записей чисел в позиционной системе счисления. Алгоритмы, связанные с делимостью целых чисел. Алгоритм Евклида для определения НОД двух натуральных чисел. Алгоритмы линейной (однопроходной) обработки последовательности чисел без использования дополнительной памяти, зависящей от длины последовательности (вычисление максимума, суммы; линейный поиск и т.п.). Обработка элементов

последовательности, удовлетворяющих определенному условию (вычисление суммы таких элементов, их максимума и т.п.). Рекурсивные алгоритмы. Возможность их записи без использования рекурсии. Вычисление элементов рекурсивной последовательности. Построение и анализ дерева рекурсивных вызовов. Алгоритмы обработки массивов. Вставка и удаление элементов в массиве. Сортировка одномерных массивов. Квадратичные алгоритмы сортировки (пример: сортировка пузырьком). Слияние двух отсортированных массивов в один без использования сортировки. Рекурсивная реализация сортировки путем слияния двух отсортированных подмассивов. Вычисление значения многочлена заданной степени в заданной точке (значения коэффициентов многочлена задаются массивом). Алгоритмы анализа символьных строк. Построение графика функции, заданной формулой, программой или таблицей значений. Алгоритмы приближенного решения уравнений. Алгоритмы приближенного вычисления длин и площадей. Приближенное вычисление площади фигуры методом Монте-Карло. Построение траекторий, заданных разностными схемами. Решение задач оптимизации. Алгоритмы вычислительной геометрии. Вероятностные алгоритмы. Сохранение и использование промежуточных результатов. Метод динамического программирования).

2. Элементы теории алгоритмов (Формализация понятия алгоритма. Машина Тьюринга – пример абстрактной универсальной вычислительной модели. Тезис Чёрча – Тьюринга. Другие универсальные вычислительные модели (пример: машина Поста). Универсальный алгоритм. Вычислимые и невычислимые функции. Проблема остановки и ее неразрешимость. Абстрактные универсальные порождающие модели (пример: грамматики). Сложность вычисления: количество выполненных операций, размер используемой памяти; их зависимость от размера исходных данных. Сложность алгоритма сортировки слиянием (MergeSort). Примеры задач анализа алгоритмов: определение входных данных, при которых алгоритм дает указанный результат; определение результата алгоритма без его полного пошагового выполнения).

Программирование

1. Языки программирования (Подпрограммы (процедуры, функции). Параметры подпрограмм. Рекурсивные процедуры и функции. Логические переменные. Символьные и строковые переменные. Операции над строками. Двумерные массивы (матрицы). Многомерные массивы. Средства работы с данными во внешней памяти. Файлы. Подробное знакомство с одним из универсальных процедурных языков программирования. Запись алгоритмических конструкций и структур данных в выбранном языке программирования. Обзор процедурных языков программирования. Понятие о не процедурных языках программирования и парадигмах программирования).
2. Разработка программ (Этапы решения задач на компьютере. Структурное программирование. Проверка условия выполнения цикла до начала выполнения тела цикла и после выполнения тела цикла: постусловие и предусловие цикла. Инвариант цикла. Методы проектирования программ «сверху-вниз» и «снизу-вверх». Разработка программ, использующих подпрограммы. Библиотеки подпрограмм и их использование. Понятие об объектно-ориентированном программировании. Объекты и классы. Инкапсуляция, наследование, полиморфизм).

2. Основные умения и навыки

Экзаменуемый должен уметь и владеть навыками:

Находить значение логических выражений. Производить тождественные преобразования логических выражений. Знания единиц измерения информации, принципов кодирования, систем счисления, понятия алгоритма, его свойств, способов записи, основных алгоритмических конструкций. Уметь и владеть навыками: анализировать однозначность двоичного кода, формировать для логической функции таблицу истинности, осуществлять преобразования логических выражений, вычислять логическое значение сложного

высказывания по известным значениям элементарных высказываний, оценивать объем памяти, необходимый для хранения информации, оценивать скорость передачи и обработки информации, искать кратчайший путь в графе, осуществлять обход графа, осуществлять перевод из одной системы счисления в другую, оперировать массивами данных, проводить вычисления в электронных таблицах, использовать стандартные алгоритмические конструкции при программировании, формально исполнять алгоритмы, записанные на естественных и алгоритмических языках, в том числе на языках программирования, анализировать результат исполнения алгоритма, формулировать запросы к базам данных и поисковым системам, анализировать текст программы с точки зрения соответствия записанного алгоритма поставленной задаче и изменять его в соответствии с заданием, реализовывать сложный алгоритм с использованием современных систем программирования, читать и отлаживать программы на языке программирования.

Примерные варианты заданий по информатике, алгоритмизации и программированию

Письменный экзамен по информатике, алгоритмизации и программированию
Контрольно-измерительный материал № 1.

Для заданий 1-5 достаточно привести ответы. Для заданий 6 - 7 необходимо привести развернутые решения!

1. Известно, что логическое выражение A принимает значение истина, а B — ложь. Какое значение принимает логическое выражение $A \vee B \wedge \neg B$. В ответе укажите 0, если принимает значение ложь, 1 — если истина, и -1, если при заданные значениях A и B однозначно сказать нельзя.
2. Сколько единиц в двоичной записи шестнадцатеричного числа 1F31?
3. Миша заполнял таблицу истинности функции $(\neg x \vee \neg y) \wedge \neg(y \equiv z) \wedge w$, но успел заполнить лишь фрагмент из трёх различных её строк, даже не указав, какому столбцу таблицы соответствует каждая из переменных w, x, y, z .

				$(\neg x \vee \neg y) \wedge \neg(y \equiv z) \wedge w$
	0	0	1	1
0		0		1
1	0		1	1

Определите, какому столбцу таблицы соответствует каждая из переменных w, x, y, z . В ответе напишите буквы w, x, y, z в том порядке, в котором идут соответствующие им столбцы (сначала буква, соответствующая первому столбцу; затем буква, соответствующая второму столбцу, и т.д.). Буквы в ответе пишите подряд, никаких разделителей между буквами ставить не нужно.

Пример. Функция задана выражением $\neg x \vee y$, зависящим от двух переменных, а фрагмент таблицы имеет следующий вид.

		$\neg x \vee y$
0	1	0

В этом случае первому столбцу соответствует переменная y , а второму

столбцу – переменная x . В ответе следует написать yx .

4. На вход алгоритма подаётся натуральное число N . Алгоритм строит по нему новое число R следующим образом.

1. Строится двоичная запись числа N.

2. К этой записи дописываются справа и слева ещё по одному или два разряда по следующему правилу: если N чётное, то в конец числа (справа) дописывается ноль, а в начало числа (слева) дописывается единица; если N нечётное, то в конец числа (справа) и в начало числа (слева) дописываются по две единицы. Например, для числа 13 двоичная запись 1101 преобразуется в запись 11110111.

Полученная таким образом запись (в ней на два или четыре разряда больше, чем в записи исходного числа N) является двоичной записью искомого числа R. Укажите наименьшее число R, превышающее 48, которое может являться результатом работы данного алгоритма. В ответе это число запишите в десятичной системе счисления.

5. Запишите число, которое будет напечатано в результате выполнения следующей программы. Для Вашего удобства программа представлена на пяти языках программирования.

<p>Бейсик</p> <pre>DIM S, N AS INTEGER S = 0 N = 90 WHILE S + N < 145 S = S + 20 N = N - 10 WEND PRINT S</pre>	<p>Python</p> <pre>s = 0 n = 90 while s + n < 145: s = s + 20 n = n - 10 print(s)</pre>
<p>Алгоритмический язык</p> <pre>алг нач цел n, s s := 0 n := 90 нц пока s + n < 145 s := s + 20 n := n - 10 кц вывод s кон</pre>	<p>Паскаль</p> <pre>var s, n: integer; begin s := 0; n := 90; while s + n < 145 do begin s := s + 20; n := n - 10 end; writeln(s) end.</pre>
<p>C++</p> <pre>#include <iostream> using namespace std; int main() { int s = 0, n = 90; while (s + n < 145) { s = s + 20; n = n - 10; } cout << s << endl; return 0; }</pre>	

6. В языке запросов поискового сервера для обозначения логической операции «ИЛИ» используется символ «|», а для обозначения логической операции «И» – символ «&». В таблице приведены запросы и количество найденных по ним страниц некоторого сегмента сети Интернет.

Запрос	Найдено страниц (в сотнях тысяч)
<i>Масло</i>	48
<i>Двигатель</i>	28
<i>Подсолнечник</i>	52
<i>Масло Двигатель Подсолнечник</i>	83
<i>Масло & Подсолнечник</i>	19
<i>Подсолнечник & Двигатель</i>	0

Какое количество страниц (в сотнях тысяч) будет найдено по запросу Двигатель & Масло? Считается, что все запросы выполнялись практически одновременно, так что набор страниц, содержащих все искомые слова, не изменялся за время выполнения запросов.

7. Два игрока, Паша и Валя, играют в следующую игру. Перед игроками лежит куча камней. Игроки ходят по очереди, первый ход делает Паша. За один ход игрок может добавить в кучу один камень или увеличить количество камней в куче в два раза. Например, имея кучу из 15 камней, за один ход можно получить кучу из 16 или 30 камней. У каждого игрока, чтобы делать ходы, есть неограниченное количество камней. Игра завершается в тот момент, когда количество камней в куче становится не менее 32. Если при этом в куче оказалось не более 54 камней, то победителем считается игрок, сделавший последний ход. В противном случае победителем становится его противник. Например, если в куче было 29 камней и Паша удвоит количество камней в куче, то игра закончится и победителем будет Валя. В начальный момент в куче было S камней, $1 \leq S \leq 31$. Будем говорить, что игрок имеет выигрышную стратегию, если он может выиграть при любых ходах противника. Описать стратегию игрока – значит описать, какой ход он должен сделать в любой ситуации, которая ему может встретиться при различной игре противника.

Выполните следующие задания.

- а) При каких значениях числа S Паша может выиграть в один ход? Укажите все такие значения и соответствующие ходы Паши.
б) У кого из игроков есть выигрышная стратегия при $S = 30, 29, 28$? Опишите выигрышные стратегии для этих случаев.
- У кого из игроков есть выигрышная стратегия при $S = 15, 14$? Опишите соответствующие выигрышные стратегии.
- У кого из игроков есть выигрышная стратегия при $S = 13$? Постройте дерево всех партий, возможных при этой выигрышной стратегии (в виде рисунка или таблицы). На рёбрах дерева указывайте, кто делает ход; в узлах – количество камней в позиции.

Критерии оценивания ответов поступающих

Оценка работы проводится в баллах. Правильное решение каждой из задач 1-5 оценивается в 10 баллов. Правильное решение задач 6 оценивается в 20 баллов, задачи 7 — в 30 балла, задачи 11 — в 30 баллов.

Итоговая оценка — сумма баллов. Максимальный балл — 100.

Длительность экзамена — 180 минут.

РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

1. Семакин И.Г., Шестакова Л.В., Хеннер Е.К.: Информатика. 11 класс. Учебник. Углубленный уровень. В 2-х частях. - Бином, Лаборатория знаний, 2020. - 176 с.
2. Поляков К.Ю., Еремин Е.А. Информатика. 10 класс. Учебник. Базовый и углубленный уровни. В 2-х частях. - Бином, Лаборатория знаний, 2020. - 352 с.
3. Поляков К.Ю., Еремин Е.А. Информатика. 11 класс. Учебник. Базовый и углубленный уровни. В 2-х частях. - Бином, Лаборатория знаний, 2020. - 304 с.
4. Гейн А.Г., Гейн А.А. Информатика. 11 класс. Учебник. Базовый уровень. - М. : Просвещение, 2019. - 128 с.
5. Гейн А.Г., Сенокосов А.И., Ливчак А.Б. Информатика. 10 класс. Учебник. Базовый и углубленный уровни. - М. : Просвещение, 2019. - 272 с.
6. Ушаков Д. М. Информатика. Сборник заданий с решениями и ответами для подготовки к ЕГЭ. - АСТ, 2019. - 528 с.
7. Крылов С.С. ЕГЭ 2020. Тренажёр. Информатика. - М. : Экзамен, 2019.