

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Усынин Максим Валерьевич

Должность: Ректор

Дата подписания: 19.11.2021 г.

Уникальный программный ключ:

f498e59e83f65dd7c3ce7bb8a25cbbabb33ebc58

**Частное образовательное учреждение высшего образования  
«Международный Институт Дизайна и Сервиса»  
(ЧОУВО МИДиС)**

Кафедра математики и информатики

**ПРОГРАММА  
ВСТУПИТЕЛЬНОГО ИСПЫТАНИЯ  
ПО ПРОГРАММИРОВАНИЮ**

Челябинск – 2021

Разработчик – преподаватель кафедры математики и информатики Е.В. Прилепина

Программа рассмотрена и одобрена на заседании кафедры математики и информатики.

Протокол № 1 от 26 августа 2021 г.

Заведующий кафедрой математики и  
информатики



Л.Ю. Овсянникова

## **1. Пояснительная записка**

Программа предназначена для абитуриентов, поступающих в ЧОУВО МИДиС на направления высшего образования: 09.03.03 Прикладная информатика. Экзаменационные задания по программированию разработаны с требованиями ФГОС специальности 09.02.07 Информационные технологии и программирование от 9 декабря 2016 г. № 1547.

Данная программа ставит целью проверки знаний, навыков и умений по программированию у поступающих в ЧОУВО МИДиС абитуриентов в объеме программы общеобразовательной школы.

Содержание заданий разработано по темам курса информатики, объединённым в тематический блоки: «Элементы теории алгоритмов», «Программирование».

Содержанием работы охватывается основное содержание блоков «Элементы теории алгоритмов» и «Программирование» курса информатики

Работа содержит как задания базового уровня сложности, проверяющие знания и умения, предусмотренные требованиями базового уровня освоения основной образовательной программы, так и задания повышенного и высокого уровней сложности, проверяющие знания и умения, предусмотренные требованиями профильного уровня.

Структура работы обеспечивает оптимальный баланс заданий разных типов и разновидностей, трёх уровней сложности, проверяющих знания и умения на трёх различных уровнях: воспроизведения, применения в стандартной ситуации, применения в новой ситуации.

На вступительном испытании по программированию абитуриент должен продемонстрировать следующие знания, умения и навыки:

- знание основных алгоритмических конструкций;
- умение записи алгоритмических конструкций в выбранном языке;
- умение решать алгоритмические задачи, связанных с анализом графов (примеры: построение оптимального пути между вершинами ориентированного ациклического графа; определение количества различных путей между вершинами);
- знание основных конструкций программирования;
- умение понимать программы, написанные на выбранном для изучения универсальном алгоритмическом языке высокого уровня;
- владение стандартными приёмами написания на алгоритмическом языке программы для решения стандартной задачи с использованием основных конструкций программирования и отладки таких программ;
- владение навыками и опытом разработки программ в среде программирования, включая тестирование и отладку программ;
- владение элементарными навыками формализации прикладной задачи и документирования программ;
- умение выполнять сортировку одномерных массивов;
- иметь понятие о квадратичных алгоритмах сортировки (пример: сортировка пузырьком);
- умение выполнять слияние двух отсортированных массивов в один без использования сортировки;
- знать алгоритмы анализа отсортированных массивов;
- умение выполнять рекурсивную реализацию сортировки массива на основе слияния двух его отсортированных фрагментов.

Каждый вариант работы включает в себя 12 заданий, различающихся уровнем сложности. Распределение заданий по разделам представлено в таблице 1.

Таблица 1  
Распределение заданий по содержательным разделам

№	Проверяемые элементы содержания	Уровень сложности
1.	Умение представлять и считывать данные в разных типах информационных моделей (схемы, карты, таблицы, графики и формулы)	Б
2.	Умение представлять и считывать данные в разных типах информационных моделей (схемы, карты, таблицы, графики и формулы)	П
3.	Формальное исполнение простого алгоритма, записанного на естественном языке, или умение создавать линейный алгоритм для формального исполнителя с ограниченным набором команд, или умение восстанавливать исходные данные линейного алгоритма по результатам его работы.	Б
4.	Знание основных конструкций языка программирования, понятия переменной, оператора присваивания	Б
5.	Умение исполнить алгоритм для конкретного исполнителя с фиксированным набором команд	Б
6.	Вычисление рекуррентных выражений	П
7.	Умение анализировать алгоритм логической игры	Б
8.	Умение найти выигрышную стратегию игры	П
9.	Умение построить дерево игры по заданному алгоритму и найти выигрышную стратегию	В
10.	Умение анализировать алгоритм, содержащий ветвление и цикл	П
11.	Умение анализировать результат исполнения алгоритма, содержащего ветвление и цикл	Б
12.	Умение анализировать результат исполнения алгоритма, содержащего ветвление и цикл	П

## 2. Содержание программы

### Элементы теории алгоритмов и программирование.

Алгоритмы исследования элементарных функций, в частности – точного и приближенного решения квадратного уравнения с целыми и вещественными коэффициентами, определения экстремумов квадратичной функции на отрезке.

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 определённому условию (вычисление суммы заданных элементов, их максимума и т.п.).

Алгоритмы обработки массивов. Примеры: перестановка элементов данного одномерного массива в обратном порядке; циклический сдвиг элементов массива; заполнение двумерного числового массива по заданным правилам; поиск элемента в двумерном массиве; вычисление максимума и суммы элементов двумерного массива. Вставка и удаление элементов в массиве.

Рекурсивные алгоритмы, в частности: нахождение натуральной и целой степени заданного ненулевого вещественного числа; вычисление факториалов; вычисление n-го элемента рекуррентной последовательности (например,

последовательности Фибоначчи). Построение и анализ дерева рекурсивных вызовов. Возможность записи рекурсивных алгоритмов без явного использования рекурсии.

Алгоритмы анализа символьных строк, в том числе: подсчёт количества появлений символа в строке; разбиение строки на слова по пробельным символам; поиск подстроки внутри данной строки; замена найденной подстроки на другую строку.

Алгоритмы приближенного решения уравнений на данном отрезке, например, методом деления отрезка пополам.

Алгоритмы приближенного вычисления длин и площадей, в том числе: приближенное вычисление длины плоской кривой путём аппроксимации её ломаной; приближённый подсчёт методом трапеций площади под графиком функции, заданной формулой, программой или таблицей значений.

### 3. Литература

1. Босова, Л.Л. Информатика. 10 класс. Базовый уровень [Текст]: учебник / Л.Л. Босова, А.Ю. Босова. – М. : БИНОМ. Лаборатория знаний, 2016. – 256 с.

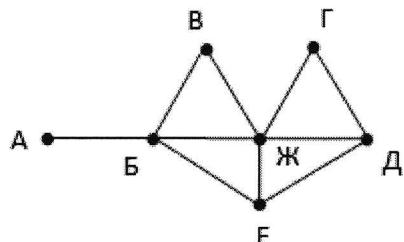
2. Босова, Л.Л. Информатика. 11 класс. Базовый уровень [Текст]: учебник / Л.Л. Босова, А.Ю. Босова. – М. : БИНОМ. Лаборатория знаний, 2016. – 256 с.

3. Угринович, Н.Д. Информатика. 10–11 класс. Базовый уровень [Текст]: учебник / Н.Д. Угринович. – М. : БИНОМ. Лаборатория знаний, 2016. – 272 с.

### 4. Демоверсия вступительного испытания

**Вопрос 1.** На рисунке справа схема дорог Н-ского района изображена в виде графа, в таблице содержатся сведения о длинах этих дорог (в километрах). Так как таблицу и схему рисовали независимо друг от друга, то нумерация населённых пунктов в таблице никак не связана с буквенными обозначениями на графике. Определите, какова протяжённость дороги из пункта Г в пункт Ж. В ответе запишите целое число – так, как оно указано в таблице.

	1	2	3	4	5	6	7
1				9			7
2			5			11	
3						12	
4	9	5			4	13	15
5				4		10	8
6		11	12	13	10		
7	7			15	8		



**Вопрос 2.** 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 ещё два разряда по следующему правилу:

а) складываются все цифры двоичной записи числа N, и остаток от деления суммы на 2 дописывается в конец числа (справа). Например, запись 11100 преобразуется в запись 111001;

б) над этой записью производятся те же действия – справа дописывается остаток от деления суммы её цифр на 2.

Полученная таким образом запись (в ней на два разряда больше, чем в записи исходного числа N) является двоичной записью искомого числа R. Укажите такое

наименьшее число  $N$ , для которого результат работы данного алгоритма больше числа 77. В ответе это число запишите в десятичной системе счисления.

**Вопрос 3.** Определите, при каком наименьшем введённом значении переменной  $s$  программа выведет число 64.

```
var s, n: integer;
begin
    readln (s);
    n := 1;
    while s < 51 do begin
        s := s + 5;
        n := n * 2
    end;
    writeln(n)
end.
```

**Вопрос 4.** Редактор получает на вход строку цифр и преобразовывает её. Редактор может выполнять две команды, в обеих командах  $v$  и  $w$  обозначают цепочки цифр.

**заменить ( $v$ ,  $w$ )**

Эта команда заменяет в строке первое слева вхождение цепочки  $v$  на цепочку  $w$ . Если цепочки  $v$  в строке нет, эта команда не изменяет строку.

**нашлось ( $v$ )**

Эта команда проверяет, встречается ли цепочка  $v$  в строке исполнителя Редактор. Если она встречается, то команда возвращает логическое значение «истина», в противном случае возвращает значение «ложь». Стока при этом не изменяется.

Дана программа для исполнителя Редактор:

**НАЧАЛО**

**ПОКА** нашлось (2222) ИЛИ нашлось (8888)

**ЕСЛИ** нашлось (2222)

**ТО** заменить (2222, 88)

**ИНАЧЕ** заменить (8888, 22)

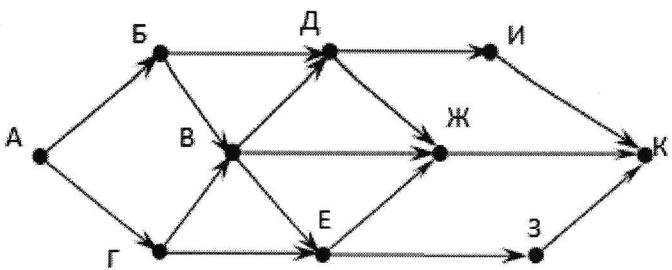
**КОНЕЦ ЕСЛИ**

**КОНЕЦ ПОКА**

**КОНЕЦ**

Какая строка получится в результате применения приведённой программы к строке, состоящей из 70 идущих подряд цифр 8? В ответе запишите полученную строку.

**Вопрос 5.** 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 маршрутов из города А в город К, содержащих ровно пять городов, включая города А и К?



**Вопрос 6.** Алгоритм вычисления функции  $F(n)$  задан следующими соотношениями:

$$\begin{aligned} F(n) &= 2n \text{ при } n \leq 5 \\ F(n) &= F(n-2) + 3 \cdot F(n/2) + n, \text{ если } n > 5 \text{ и чётно,} \\ F(n) &= F(n-1) + F(n-2) + F(n-3), \text{ если } n > 5 \text{ и нечётно.} \end{aligned}$$

Чему равно значение функции  $F(5) + F(6)$ ?

**Вопрос 7.** Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежит куча камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может добавить в кучу один камень или увеличить количество камней в куче в два раза. Для того чтобы делать ходы, у каждого игрока есть неограниченное количество камней. Игра завершается в тот момент, когда количество камней в куче становится не менее 29. Победителем считается игрок, сделавший последний ход, т.е. первым получивший кучу, в которой будет 29 или больше камней.

В начальный момент в куче было  $S$  камней,  $1 \leq S \leq 28$ .

Укажите такое значение  $S$ , при котором Петя не может выиграть за один ход, но при любом ходе Пети Ваня может выиграть своим первым ходом.

**Вопрос 8.** Найдите два таких значения  $S$ , при которых у Пети есть выигрышная стратегия, причём одновременно выполняются два условия:

- Петя не может выиграть за один ход;
- Петя может выиграть своим вторым ходом независимо от того, как будет ходить Ваня.

Найденные значения запишите в ответе в порядке возрастания, через пробел.

**Вопрос 9.** Найдите значение  $S$ , при котором одновременно выполняются два условия:

- у Вани есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Пети;
- у Вани нет стратегии, которая позволит ему гарантированно выиграть первым ходом.

**Вопрос 10.** Получив на вход число  $x$ , этот алгоритм печатает два числа:  $L$  и  $M$ . Укажите наибольшее число  $x$ , при вводе которого алгоритм печатает сначала 4, а потом 5.

```
var x, L, M, Q: integer;
begin
  readln(x);
```

```

Q := 9;
L := 0;
while x >= Q do begin
    L := L + 1;
    x := x - Q;
end;
M := x;
if M < L then begin
    M := L;
    L := x;
end;
writeln(L);
writeln(M);
end.

```

**Вопрос 11.** Исполнитель преобразует число на экране. У исполнителя есть две команды, которым присвоены номера:

1. Прибавить 1
2. Умножить на 2

Первая команда увеличивает число на экране на 1, вторая умножает его на 2. Программа для исполнителя – это последовательность команд. Сколько существует программ, для которых при исходном числе 1 результатом является число 20, и при этом траектория вычислений содержит число 10?

**Вопрос 12.** У исполнителя Калькулятор две команды:

1. прибавь 2
2. умножь на 3.

Первая из них увеличивает число на экране на 2, вторая — утраивает его. Сколько различных чисел можно получить из числа 2 с помощью программы, которая содержит ровно 3 команды?

#### Правильные ответы

№ задания	Ответ
1.	9
2.	19
3.	21
4.	22
5.	6
6.	30
7.	14
8.	7 13
9.	12
10.	49
11.	28
12.	8

## **5. Критерии оценивания**

В тесте 12 вопросов. Оценка 3 выставляется при выполнении от 30 до 60 % заданий, оценка 4 от 61 до 90 %, оценка 5 от 91 до 100 %.

## **6. Инструкция проведения вступительного испытания**

Вступительное испытание проходит в форме компьютерного тестирования. Тест содержит 12 вопросов со свободным ответом. Необходимо написать правильный ответ.

Тестирование проходит по расписанию. Расписание утверждается к 1 июня и размещается на сайте ЧОУВО МИДиС и информационном стенде.

Пропуском на экзамен является распечатка Логина и пароля в личный кабинет и Паспорт.

Строго запрещено пользоваться мобильным телефоном, планшетами и другими подобными устройствами. При обнаружении организаторами этих устройств абитуриент удаляется с экзамена без права пересдачи.

Время выполнения теста – 60 мин.