

ПРОГРАММА ВСТУПИТЕЛЬНОГО ИСПЫТАНИЯ ПО ДИСЦИПЛИНЕ «ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»

(для выпускников учреждений среднего специального образования по специальностям «Программируемые мобильные системы», «Программное обеспечение информационных технологий», «Вычислительные машины, системы и сети», «Тестирование программного обеспечения», «Электронные вычислительные средства»)

1. ОБЩИЕ УКАЗАНИЯ

Программа профильного вступительного испытания по дисциплине «Основы алгоритмизации и программирования» рассчитана на поступающих в 2018 году на механико-математический факультет БГУ для получения высшего образования I ступени по специальности 1-31 03 08 «Математика и информационные технологии» (сокращенная форма).

Цель проведения вступительного испытания по дисциплине «Основы алгоритмизации и программирования» – оценить уровень теоретической и практической подготовки абитуриентов в разработке алгоритмов и программ на процедурно-ориентированном языке программирования при решении задач.

Задачи проведения вступительного испытания по дисциплине «Основы алгоритмизации и программирования»:

– выявить необходимые знания основ алгоритмизации, методов разработки программ, основных элементов и принципов программирования на процедурно-ориентированном языке программирования;

– оценить навыки разработки алгоритмов и программ на одном из процедурных языков программирования (например, Паскаль или Си).

В результате прохождения вступительных испытаний абитуриенты должны продемонстрировать **знания:**

– понятия алгоритма, свойств алгоритмов, общих принципов построения алгоритмов, способов описания алгоритмов, типов алгоритмов;

– принципов построения эффективных алгоритмов;

– методов разработки программ, основных элементов языка программирования, операторов, функций и операций, управляющих структур, структур данных, файлов;

– принципов программирования на одном из алгоритмических языков высокого уровня (как, например, Pascal, или C, или C++, или Java, или т. п.);

умения:

– разрабатывать алгоритм решения задачи;

– разрабатывать программы, проводить их отладку, тестирование и верификацию.

Вступительное испытание проводится в форме письменного экзамена. Абитуриентам предлагается решить **10 задач за 120 минут**.

2. ТРЕБОВАНИЯ К ВСТУПИТЕЛЬНОМУ ИСПЫТАНИЮ

2.1 Содержание программы вступительного испытания

ТЕМА 1. ОБЩИЕ СВЕДЕНИЯ ОБ АЛГОРИТМАХ

Алгоритм и его свойства. Способы описания алгоритмов. Стандартизация графического представления алгоритмов. Методы разработки и анализа алгоритмов. Псевдокоды. Проблемы создания алгоритмов. Проблема универсального языка программирования. Проблема универсальной вычислительной машины. Машина Тьюринга. Написание программ на машине Тьюринга.

ТЕМА 2. ОСНОВНЫЕ ЭЛЕМЕНТЫ ЯЗЫКА

Основные понятия языка. Структура программы. Простые типы данных. Операции и их приоритет. Выражения. Основные операторы. Основные возможности организации ввода/вывода. Стандартные потоки ввода/вывода.

ТЕМА 3. ТИПЫ ДАННЫХ

Представление чисел в памяти компьютера. Позиционные системы счисления. Перевод чисел из одной системы счисления в другую. Массивы. Работа с массивами. Строки. Работа со строками. Структуры данных различного типа. Работа со структурами. Адреса и указатели. Основные возможности работы с динамической памятью. Задачи поиска и сортировки. Бинарный и интерполяционный поиск. Сортировка методом «пузырька», вставками, подсчетом, слиянием.

ТЕМА 4. ПОДПРОГРАММЫ

Модульность в программировании. Понятие и структура подпрограммы. Описание подпрограмм в языках высокого уровня (процедуры, функции). Организация вызова подпрограммы. Типы параметров подпрограммы (функции), локальные и глобальные переменные. Организация многофункциональных программ. Внешние модули. Рекурсивные алгоритмы и функции.

ТЕМА 5. ДИНАМИЧЕСКИЕ СТРУКТУРЫ ДАННЫХ

Организация динамических структур данных. Списки. Стеки. Очереди. Организация данных в виде древовидных динамических структур. Двоичные деревья. Деревья бинарного поиска и основные операции для работы с ними:

поиск, вставка, удаление. Принципы хеширования. Хеширование с открытой и закрытой адресацией.

ТЕМА 6. ДОПОЛНИТЕЛЬНЫЕ ВОЗМОЖНОСТИ ИЗУЧАЕМОГО ЯЗЫКА

Специфические операторы изучаемого языка программирования. Специфические типы данных. Специфические возможности изучаемого языка. Базовые принципы объектно-ориентированного программирования: инкапсуляция, наследование, полиморфизм. Понятие объекта, класса, метода, свойства. Конструкторы, деструкторы.

2.2. Примеры экзаменационных заданий

Задача 1. Дано: $a = 70_{10}$, $b = 100_8$. Какое из чисел c , записанных в двоичной системе, удовлетворяет условию $b < c < a$?

1) 1000000_2

2) 1000110_2

3) 1000101_2

4) 1000111_2

5) никакое из вышеперечисленных.

Задача 2. Алгоритм вычисления значения функции $f(n)$, где n – натуральное число, при $n \geq 2$ задан соотношением $f(n) = f(n-1) + 3f(n-2)$. Известно также, что $f(1) = 0$ и $f(2) = 2$. Найдите $f(6)$.

Задача 3. Сколько оперативной памяти должен занимать тип `type`, чтобы в результате следующего фрагмента кода переменная c хранила -32767 ?

```
type a = 32767;
```

```
type b = 1;
```

```
type c = 1;
```

```
c = a + b;
```

Задача 4. Опишите на одном из языков программирования алгоритм нахождения узла в дереве бинарного поиска, имеющего заданный ключ (числовое значение).

Задача 5. Определите значение переменной c после выполнения следующего фрагмента программы:

```
a = 17;
```

```
b = 20;
```

```
a = 3 * a - b;
```

```
if (a > b)
```

```
    c = 5 * a - b;
```

```
else
```

```
    c = 5 * a + b;
```

Задача 6. Напишите программу для машины Тьюринга, реализующую следующее преобразование.

Вход: массив из 0 и 1.

Выход: ответ «да» (1), если число 0 и число 1 совпадает; «нет» (0) – в противном случае.

Задача 7. В программе описан одномерный целочисленный массив с индексами от 0 до 10. Ниже представлен фрагмент программы, обрабатывающей данный массив:

```
s = 0;
n = 10;
for (i = 0; i <= n-3; i++)
    s=s+A[i] - A[i+2];
```

В начале выполнения этого фрагмента в массиве находились трехзначные натуральные числа. Какое наибольшее значение может иметь переменная s после выполнения данной программы?

Задача 8. Опишите на одном из языков программирования алгоритм подсчета максимального количества подряд идущих четных элементов в целочисленном массиве длины 40.

Задача 9. Дан следующий фрагмент кода:

```
int main()
{
    int a = 1, b = 1;
    int x, y;
    scanf("%d%d", &x, &y); // пользователь вводит два числа
    printf("%d %d", a, b);
    for(int i = 3; i <= 10; ++i)
    {
        int c = x * a + y * b;
        printf("%d ", c);
        a = b;
        b = c;
    }
    printf("\n");
    return 0;
}
```

Каков будет результат, если пользователь введет числа 1 и 2?

Задача 10. В левом нижнем углу клетчатого поля

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

стоит робот. Из каждой клетки он может пойти только вправо или вверх. Определите, какова максимальная сумма чисел, которую робот сможет набрать по пути в правый верхний угол?

3. ОЦЕНКА РЕЗУЛЬТАТОВ ВСТУПИТЕЛЬНОГО ИСПЫТАНИЯ

Все задачи, кроме тестовых (задача 1), предполагают наличие решения с пояснениями. Если абитуриент приводит лишь ответ к задаче, то она оценивается в 0 баллов вне зависимости от правильности ответа.

Определение оценки экзаменационной работы осуществляется в два этапа:

1) подсчет количества баллов, фактически набранных абитуриентом, за выполнение каждого задания в соответствии с таблицей 1;

2) перевод суммарного количества баллов в оценку работы абитуриента в соответствии с таблицей 2.

Таблица 1

№ задачи	1	2	3	4	5	6	7	8	9	10	Сумма
Балл	1	1	2	2	2	2	3	3	4	4	24

Максимальное количество баллов за выполнение каждого задания

Таблица 2

Число баллов	0	1	2	3–5	6–8	9–11	12–15	16–19	20–22	23–24
Оценка	1	2	3	4	5	6	7	8	9	10

Шкала перевода суммарного количества баллов в оценку работы абитуриента

В случае, когда абитуриент присутствовал на вступительном испытании, но экзаменационную работу не выполнял, ему выставляется оценка 1.

Каждое задание может быть:

- выполненным;
- выполненным частично;
- невыполненным.

Задание считается **выполненным**, если ответ абитуриента удовлетворяет следующим критериям:

- правильно выбран способ решения задачи;
- в коде программы допущено не более двух синтаксических ошибок;
- код программы работает во всех требуемых случаях;
- даны точные определения необходимых терминов;
- временная сложность программы удовлетворяет требованиям условия задачи;
- получен правильный ответ.

Задание считается **выполненным частично**, если способ решения задачи выбран правильно, однако в коде программы допущено не менее трех и не более восьми синтаксических ошибок; код программы работает в более чем 75%

возможных случаев; временная сложность программы не удовлетворяет требованиям условия задачи.

Задание считается **невыполненным**, если способ решения задачи выбран неправильно; в коде программы допущено более восьми синтаксических ошибок; код программы работает в менее чем 75% возможных случаев.

Оценка за каждое задание должна равняться целому числу баллов.

Если задание выполнено, абитуриент получает максимально возможное число баллов.

Если задание выполнено частично, то абитуриент получает за него не более 75% баллов от максимально возможного числа баллов. Если при этом оценка не является целым числом, то производится округление до ближайшего целого в меньшую сторону.

За невыполненное задание абитуриент получает 0 баллов.

4. РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

ОСНОВНАЯ

1. *Вирт, Н.* Алгоритмы и структуры данных / Н. Вирт. – Санкт-Петербург: Невский диалект, 2001. – 352 с.
2. *Кормен, Т.Х.* Алгоритмы. Построение и анализ / Т.Х. Кормен., Ч.И. Лейзерсон, Р.Л. Ривест, К. Штайн. – М.: Вильямс, 2005. – 1296 с.
3. *Шень, А.Х.* Программирование: теоремы и задачи / А.Х. Шень. – М.: МЦНМО, 2004. – 296 с.

ДОПОЛНИТЕЛЬНАЯ

4. *Ахо, А.* Построение и анализ вычислительных алгоритмов / А. Ахо, Дж. Хопкрофт, Дж. Ульман. – М.: Издательский дом «Вильямс», 2001. – 384 с.
5. *Кнут, Д.* Искусство программирования. Т. 1. Основные алгоритмы / Д. Кнут. – М.: Вильямс, 2006. – 720 с.
6. *Кнут, Д.* Искусство программирования. Т. 2. Получисленные алгоритмы / Д. Кнут. – М.: Вильямс, 2007. – 832 с.
7. *Кнут, Д.* Искусство программирования. Т. 3. Сортировка и поиск / Д. Кнут. – М.: Вильямс, 2007. – 824 с.