

Министерство образования и науки Мурманской области

Государственное автономное профессиональное образовательное учреждение  
Мурманской области «Мурманский колледж экономики и информационных технологий»

Центр цифрового образования детей «IT – куб»

РАССМОТРЕНА  
Методическим советом  
ГАПОУ МО «МКЭиИТ»  
Протокол № 6 от 26.05.2023 г.



УТВЕРЖДЕНА  
Приказом директора  
ГАПОУ МО «МКЭиИТ»  
№ 75 от «09» июня 2023 г.

**Дополнительная общеобразовательная  
общеразвивающая программа технической  
направленности  
«Введение в олимпиадное программирование»**

Направленность: техническая

Уровень программы: продвинутый

Срок реализации: 1 год (72 часа)

Возраст детей: 13 - 17 лет

**Разработчик:**  
Кузнецова К.В., педагог дополнительного  
образования, методист ЦЦОД «IT-куб»

г. Мурманск

2023 г.

## Содержание

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА .....	3
УЧЕБНЫЙ ПЛАН .....	12
СОДЕРЖАНИЕ УЧЕБНОГО ПЛАНА .....	15
КОМПЛЕКС ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИХ УСЛОВИЙ.....	19
СПИСОК ЛИТЕРАТУРЫ .....	22
Приложение 1.....	23
Приложение 2.....	27

## ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

### 1. Нормативно-правовая база разработки и реализации программы.

Программа разработана на основе следующих нормативных документов:

- Федерального закона от 29.12.2012 № 273-ФЗ (ред. от 31.07.2020) «Об образовании в Российской Федерации» (с изм. и доп., вступ. в силу с 01.09.2020);
- Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам, утвержденного приказом Министерства просвещения Российской Федерации от 27.07.2022 г. № 629;
- Постановления Главного государственного санитарного врача Российской Федерации от 28.09.2020 № 28 «Об утверждении санитарных правил СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи»;
- Постановления Главного государственного санитарного врача Российской Федерации от 28.01.2021 № 2 «Об утверждении санитарных правил и норм СанПиН 1.2.3685-21 Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания»;
- Приказа Минобрнауки России №882, Минпросвещения России №391 от 05.08.2020 «Об организации и осуществлении образовательной деятельности при сетевой форме реализации образовательных программ» (вместе с «Порядком организации и осуществления образовательной деятельности при сетевой форме реализации образовательных программ»);
- Методических рекомендаций по созданию и функционированию центров цифрового образования «IT-куб» (письмо Министерства просвещения Российской Федерации от 10 ноября 2021 г. № ТВ-1984/04);
- Стратегии развития воспитания в Российской Федерации на период до 2025 года (Утверждена распоряжением Правительства РФ от 29.05.2015 № 996-р «Об утверждении Стратегии развития воспитания в Российской Федерации на период до 2025 года»);

– Концепции развития дополнительного образования детей до 2030 года, утвержденной распоряжением Правительства Российской Федерации от 31 марта 2022 г. № 678-р.

## **2. Новизна, актуальность, педагогическая целесообразность реализации программы.**

Актуальность программы обусловлена возросшей потребностью в отечественном программном обеспечении и увеличением числа программируемой техники в различных сферах жизни человека. В основе любой программы лежит алгоритм, которых может быть реализован на любом языке программирования. В условиях постоянно растущего объема информации оптимизация алгоритмов становится важной задачей при программировании в любой отрасли. Большинство олимпиад по программированию содержит задачи, которые можно решить только при эффективном использовании ресурсов участника и компьютера.

Программу отличает новизна содержания. В процессе освоения программы обучающиеся осваивают второй язык программирования: C++ или Python для наиболее эффективного решения олимпиадных задач.

Педагогическая целесообразность программы определяется образовательным замыслом. В первой половине года обучающиеся изучают второй язык программирования, а во второй половине года изучают распространенные в олимпиадных задачах алгоритмы. В процессе реализации данной программы предусмотрено участие обучающихся в олимпиадах по программированию.

**Направленность программы:** техническая.

## **3. Адресат программы.**

Адресатом программы являются дети в возрасте от 13 до 17 лет.

Содержание и объем стартовых знаний, необходимых для начального этапа освоения программы: базовые навыки работы с текстовыми редакторами, браузерами, поисковыми системами, файловыми менеджерами (проводником), знание одного из языков программирования: Python или C++.

#### **4. Срок реализации программы.**

Срок реализации программы составляет 1 год.

**Уровень программы:** продвинутый.

Программа предполагает использование форм организации материала, обеспечивающих доступ к сложным (возможно узкоспециализированным) и нетривиальным разделам в рамках содержательно-тематического направления программы. Также предполагает углубленное изучение содержания программы и доступ к около профессиональным и профессиональным знаниям в рамках содержательно-тематического направления программы.

#### **5. Форма реализации программы.**

Форма обучения – очная.

Образовательные технологии: информационные технологии, проектная технология, здоровьесберегающие технологии, технология проблемного обучения.

Форма организации содержания и процесса педагогической деятельности – комплексная.

Тип организации работы учеников: групповая работа, индивидуальная, коллективная.

Виды занятий: лекции и практические занятия.

**Наполняемость группы:** от 10 до 12 человек.

#### **6. Объём программы и режим работы**

**Объём программы:** 72 часа.

**Режим занятий:** 1 - 2 раза в неделю по 2 академических часа.

Продолжительность часа – 40 минут.

#### **7. Цель программы:**

Целью программы является создание условий для приобретения перспективного опыта реализации алгоритмов обработки данных на C++ и Python и участия в олимпиадах по информатике.

## **8. Задачи программы.**

Программа направлена на решение следующих задач:

- познакомить с правилами безопасного использования цифровых инструментов и компьютерного оборудования, организации рабочего места;
- изучить основы языков C++ и Python, достаточных для участия в олимпиадах по программированию;
- познакомить с различиями языков программирования C++ и Python;
- познакомить с часто встречающимися в олимпиадах алгоритмах;
- способствовать формированию алгоритмического и логического мышления;
- способствовать формированию познавательных и регулятивных универсальных учебных действий;
- совершенствовать навык поиска информации в сети Интернет, анализа выбранной информации на соответствие запросу, использования информации при решении задач;
- формировать коммуникативные навыки;
- развивать рефлексивную деятельность учащихся;
- воспитывать трудолюбие, целеустремленность, уважение к труду.

## **9. Планируемые результаты освоения программы**

В результате освоения программы, учащиеся будут:

### **Знать:**

- правила техники безопасности в компьютерном кабинете;
- отличия языков программирования Python и C++;
- алгоритмы сортировок, поиска, нахождения НОД и НОК, поиска суммы на отрезке.

**Уметь:**

- соблюдать технику безопасности;
- определять способ решения задачи;
- анализировать условие задачи и предлагать пути решения.

**Владеть:**

- приемами поиска информации в сети интернет.

**Метапредметные результаты:**

- выполнять основные логические действия (анализ, синтез, установление причинно-следственных связей);
- объяснять решение и аргументировать свою точку зрения;
- уметь сопоставлять полученный и ожидаемый результат, исправлять ошибки.

**Личностные результаты:**

- развитие усидчивости и стремления доводить решение до конца;
- формирование коммуникативной компетентности в общении и сотрудничестве со сверстниками.

**10. Формы представления результатов**

Формы аттестации: проверочная работа, демонстрация решения, наблюдение.

Входной контроль осуществляется в начале реализации программы в форме наблюдения и имеет диагностические задачи. Цель входной диагностики – зафиксировать начальный уровень подготовки обучающихся, имеющиеся знания, умения и навыки, связанные с предстоящей деятельностью.

Наблюдение осуществляется в течение реализации программы.

Промежуточный контроль осуществляется в целях диагностики теоретических знаний и практических умений и навыков по итогам освоения одного из разделов курса. Проводится в форме проверочной работы.

Итоговый контроль проводится с целью определения степени достижения результатов обучения и получения сведений для совершенствования программы и методов обучения – представляет из себя решение итоговой проверочной работы.

## **11. Оценочные материалы, формирующие систему оценивания**

**Промежуточная проверочная работа** состоит из 5 задач, решение которых проверяется автоматически. Примеры задач приведены в приложении 1. Максимальный балл за промежуточную аттестацию: 100 баллов.

Оценивание тестирования осуществляется по следующим уровням:  
высокий уровень – учащийся набрал не менее 80% от максимально возможного количества баллов (от 80 баллов).

средний уровень – учащийся набрал не менее 50% от максимально возможного количества баллов (от 50 баллов).

низкий уровень – учащийся набрал менее 50% от максимально возможного количества баллов (менее 50 баллов).

**Итоговое тестирование** состоит из 5 задач на программирование с проверкой в автоматической системе. Примеры заданий итоговой проверочной работы приведены в приложении 2. Максимальный балл за итоговое тестирование: 100 баллов.

Оценивание тестирования осуществляется по следующим уровням:  
высокий уровень – учащийся набрал не менее 80% от максимально возможного количества баллов (от 80 баллов).

средний уровень – учащийся набрал не менее 50% от максимально возможного количества баллов (от 50 баллов).

низкий уровень – учащийся набрал менее 50% от максимально возможного количества баллов (менее 50 баллов).

**Общими критериями оценки результативности обучения являются:**



- оценка уровня теоретических знаний: свобода восприятия теоретической информации, осмысленность и свобода использования специальной терминологии, свобода ориентации в теоретическом материале;

- оценка уровня практической подготовки: соответствие развития уровня практических умений и навыков программным требованиям, свобода владения специальным оборудованием и оснащением;

- оценка уровня достижения личностных результатов: культура организации самостоятельной деятельности, культура работы с информацией, аккуратность и ответственность при работе.

Оценка итоговых результатов освоения программы осуществляется по трем уровням:

Высокий уровень – достижение 80- 100% показателей освоения программы.

Средний уровень – достижение 50- 79% показателей освоения программы.

Низкий уровень - достижение менее чем 50% показателей освоения программы.

Достигнутые обучающимся знания, умения и навыки заносятся в сводную таблицу результатов обучения.

### Оценка уровней освоения программы

Уровни	Параметры	Показатели
<b>Высокий уровень (80-100%)</b>	Теоретические знания	Обучающийся освоил материал в полном объеме. Знает и понимает значение терминов, употребляет их осознанно и в полном соответствии с содержанием. Самостоятельно ориентируется в содержании материала по темам.
	Практические умения и навыки	Обучающийся овладел 80-100% умений и навыков, предусмотренных программой за конкретный период. Умет работать самостоятельно, применяя практические умения и навыки. Правильно и по назначению применяет инструменты. Умеет выполнять основные логические действия (анализ, синтез, установление причинно-следственных связей). Способен планировать и регулировать свою деятельность по реализации проекта. Умеет осуществлять поиск информации, в том числе в сети Интернет; выслушивать собеседника и вести диалог; выражать свои мысли в соответствии с задачами и условиями коммуникации.

	Личностные результаты	<p>Обучающийся обладает внутренней мотивацией.</p> <p>Способен самостоятельно организовать собственную деятельность.</p> <p>Сформирована культура работы с информацией.</p> <p>Работу выполняет аккуратно, доводит до конца.</p> <p>Может оценить результаты выполнения своего задания и дать оценку работы своего товарища.</p>
<b>Средний уровень (50-79%)</b>	Теоретические знания	<p>Учащийся освоил базовые знания, ориентируется в содержании материала по темам, иногда обращается за помощью к педагогу.</p> <p>Использует специальную терминологию, однако сочетает её с бытовой</p>
	Практические умения и навыки	<p>Владеет базовыми навыками и умениями, но не всегда может выполнить задание самостоятельно, просит помощи педагога.</p> <p>В основном выполняет задания на основе образца.</p> <p>Способен разрабатывать алгоритм решения при помощи преподавателя.</p> <p>Делает ошибки в работе, но может устранить их после наводящих вопросов или самостоятельно</p> <p>Испытывает незначительные затруднения при выполнении основных логических действий (анализ, синтез, установление причинно-следственных связей).</p> <p>Способен планировать и регулировать свою деятельность по реализации проекта с помощью педагога.</p> <p>Испытывает незначительные сложности в осуществлении коммуникации с педагогом и сверстниками.</p>
	Личностные результаты	<p>Внутренняя мотивация к обучению сочетается с внешней.</p> <p>В работе допускает небрежность.</p> <p>Работу не всегда выполняет аккуратно и/или доводит до конца.</p> <p>Оценить результаты своей деятельности может с подсказкой педагога.</p>
<b>Низкий уровень (меньше 50%)</b>	Теоретические знания	<p>Владеет минимальными знаниями, ориентируется в содержании материала по темам только с помощью педагога.</p> <p>Избегает употреблять специальные термины.</p>
	Практические умения и навыки	<p>Владеет минимальными начальными навыками и умениями.</p> <p>Учащийся способен выполнять каждую операцию только с подсказкой педагога или товарищей.</p> <p>В работе допускает грубые ошибки, не может их найти даже после указания преподавателя.</p> <p>В состоянии выполнять лишь простейшие практические задания педагога.</p> <p>Испытывает существенные затруднения при выполнении основных логических действий (анализ, синтез, установление причинно-следственных связей).</p> <p>Не способен планировать и регулировать свою деятельность по реализации проекта.</p> <p>Испытывает значительные сложности в осуществлении коммуникации с педагогом и сверстниками.</p>

	Личностные результаты	Преобладает внешняя мотивация к обучению. Работу часто выполняет неаккуратно и/или не доводит до конца. Не способен самостоятельно и объективно оценить результаты своей работы.
--	-----------------------	--

### **Сводная таблица результатов обучения по программе**

№ п/п	Фамилия, имя обучающегося	Оценка теоретических знаний	Оценка практических умений и навыков (предметных и метапредметных)	Личностные результаты	Итоговая оценка
----------	------------------------------	-----------------------------------	--	--------------------------	--------------------

## УЧЕБНЫЙ ПЛАН

№ п/п	Название разделов, тем	Количество часов			Форма аттестации/ контроля
		Всего	Теория	Практика	
1.	Вводное занятие: знакомство, техника безопасности. Виды олимпиад. Различия C++ и Python	2	1	1	Наблюдение
2.	Структура программы на Python и на C++	2	1	1	
3.	Типы данных, переменные, целочисленная арифметика в Python и C++	2	1	1	
4.	Математические функции в Python и C++	2	1	1	
5.	Условный оператор в Python и C++	2	1	1	Демонстрация решений
6.	Оператор множественного выбора в Python и C++	2	1	1	Демонстрация решений
7.	Виды циклов. Операторы циклов в Python и C++	2	1	1	Демонстрация решений
8.	Одномерные массивы в Python и C++	2	1	1	Демонстрация решений
9.	Двумерные массивы в Python и C++	4	1	3	Демонстрация решений
10.	Работа со строками в Python и C++	4	1	3	Демонстрация решений
11.	Функции, рекурсия в Python и C++. Передача параметров по значению и по ссылке.	4	1	3	Демонстрация решений

12.	Словари и множества в Python и C++	4	1	3	Демонстрация решений
13.	Промежуточная аттестация по пройденным темам	2	0	2	Проверочная работа
14.	Битовые операции	2	1	1	
15.	Работа с остатками. Свойства остатков. Округление.	2	1	1	Демонстрация решений
16.	Простые числа, нахождение делителей	2	1	1	Демонстрация решений
17.	Нахождение НОД и НОК	2	1	1	Демонстрация решений
18.	Быстрое возведение в степень. Нахождение факториала. Комбинаторика.	2	1	1	Демонстрация решений
19.	Алгоритмы поиска (линейный, бинарный)	2	1	1	Демонстрация решений
20.	Алгоритмы сортировок	4	1	3	Демонстрация решений
21.	Элементы с максимальной разностью, запрос суммы на отрезке, метод двух указателей	4	1	3	Демонстрация решений
22.	Динамическое программирование	4	1	3	Демонстрация решений
23.	Линейные структуры данных	4	1	3	Демонстрация решений
24.	Решение констестов олимпиад различного уровня	8	0	8	Демонстрация решений

25.	Итоговая аттестация	2	0	2	Проверочная работа
	<b>Итого</b>	<b>72</b>	<b>22</b>	<b>50</b>	

## СОДЕРЖАНИЕ УЧЕБНОГО ПЛАНА

### **1. Вводное занятие: знакомство, техника безопасности. Виды олимпиад.**

#### **Различия C++ и Python.**

Теория (1 час): виды олимпиад по программированию, различия в правилах, подходах к оценке. Различия языков программирования Python и C++.

Практика (1 час): составление индивидуальных планов участия в олимпиадах.

### **2. Структура программы на Python и на C++**

Теория (1 час): структура программы на Python и C++. Операторы ввода и вывода.

Практика (1 час): сравнение программ на ввод и вывод в Python и C++.

### **3. Типы данных, переменные, целочисленная арифметика в Python и C++**

Теория (1 час): стандартные типы данных в Python и C++. Объявление переменных. Особенности целочисленной арифметики в Python и C++.

Практика (1 час): решение задач с использованием целочисленной арифметики.

### **4. Математические функции в Python и C++**

Теория (1 час): математические функции в Python и C++.

Практика (1 часа): решение задач с использованием математических функций.

### **5. Условный оператор в Python и C++**

Теория (1 час): операторы условного выбора в Python и C++.

Практика (1 час): решение задач с использованием ветвления.

### **6. Оператор множественного выбора в Python и C++**

Теория (1 час): операторы множественного выбора в Python и C++.

Практика (1 час): решение задач с использованием операторов множественного выбора.

### **7. Виды циклов. Операторы циклов в Python и C++**

Теория (1 час): виды циклов. Операторы циклов в Python и C++.

Практика (1 час): решение задач с использованием операторов циклов.

### **8. Одномерные массивы в Python и C++**

Теория (1 час): одномерные массивы в Python и C++. Заполнение и вывод массивов.

Практика (1 час): решение задач с использованием одномерных массивов.

### **9. Двумерные массивы в Python и C++**

Теория (1 час): двумерные массивы в Python и C++.

Практика (3 час): решение задач с использованием двумерных массивов.

### **10. Работа со строками в Python и C++**

Теория (1 час): работа со строками Python и C++.

Практика (3 час): решение задач с использованием строковых данных.

### **11. Функции, рекурсия в Python и C++. Передача параметров по значению и по ссылке.**

Теория (1 час): функции, рекурсии Python и C++. Глобальные и локальные переменные, передача параметров по значению и по ссылке.

Практика (3 час): решение задач с использованием функций и рекурсии.

### **12. Словари и множества в Python и C++**

Теория (1 час): словари и множества в Python и C++.

Практика (3 час): решение задач с использованием словарей множеств.

### **13. Промежуточная аттестация по пройденным темам**

Практика (2 час): проверочная работа.

### **14. Битовые операции**

Теория (1 час): виды битовых операций и их применение при решении задач.

Практика (1 час): решение задач с использованием битовых операций.

### **15. Работа с остатками. Свойства остатков. Округление.**

Теория (1 час): остатки. Свойства остатков. Правила округления в стандартных библиотеках Python и C++.



Практика (1 час): решение задач с использованием остатков, реализация функций округления.

### **16. Простые числа, нахождение делителей**

Теория (1 час): алгоритмы проверки числа на простоту, поиск простых чисел, факторизация числа, нахождение количества делителей.

Практика (1 часа): решение задач с простыми числами.

### **17. Нахождение НОД и НОК**

Теория (1 час): определения НОД и НОК. Алгоритмы нахождения НОД и НОК.

Практика (1 часа): решение задач с использованием алгоритмов поиска НОД и НОК.

### **18. Быстрое возведение в степень. Нахождение факториала. Комбинаторика.**

Теория (1 час): алгоритм быстрого возведения в степень. Принцип «Разделяй и властвуй». Алгоритм нахождения факториала

Практика (1 часа): решение задач с использованием алгоритмов нахождения факториала (комбинаторные задачи), алгоритма быстрого возведения в степень.

### **19. Алгоритмы поиска (линейный, бинарный)**

Теория (1 час): алгоритмы линейного и бинарного поиска.

Практика (1 часа): решение задач с использованием линейного и бинарного поиска.

### **20. Алгоритмы сортировок**

Теория (1 час): стандартные инструменты для сортировки массива. Алгоритм сортировки пузырьком, сортировки вставками, сортировки выбором, сортировки слиянием, сортировка подсчетом. Алгоритм бинарного поиска.

Практика (3 часа): сравнение алгоритмов сортировок. Решение задач с использованием алгоритмов сортировок и алгоритма бинарного поиска.

### **21. Элементы с максимальной разностью, запрос суммы на отрезке, метод двух указателей**

Теория (1 час): алгоритмы поиска отрезков с наибольшей/наименьшей суммой/разностью. Частичные суммы. Метод двух указателей.

Практика (3 часа): решение задач с использованием линейных алгоритмов для обработки массивов.

## **22. Динамическое программирование**

Теория (1 час): принцип динамического программирования. Примеры решения задач с использованием динамического программирования.

Практика (3 часа): решение задач с использованием динамического программирования.

## **23. Линейные структуры данных**

Теория (1 час): принципы работы стека, очереди и дека. Варианты реализации.

Практика (3 часа): решение задач с использованием стеков, очередей и деков.

## **24. Решение контестов олимпиад различного уровня**

Практика (8 часа): решение задач из различных олимпиад прошлых лет.

## **25. Итоговая аттестация**

Практика (2 часа): проверочная работа.

# КОМПЛЕКС ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИХ УСЛОВИЙ

## Материально-техническое обеспечение

Комплекс условий реализации программы:

Аппаратное и техническое обеспечение:

а) Рабочее место учащегося

- ноутбук с выходом в сеть Интернет;

б) Рабочее место наставника

- ноутбук с выходом в сеть Интернет;

- технические средства обучения (мультимедийное устройство).

## Методическое обеспечение программы

Методы обучения:

- объяснительно-иллюстративный (беседы, объяснения);

- репродуктивный (деятельность обучаемых носит алгоритмический характер, выполняется по инструкциям, предписаниям, правилам в аналогичных, сходных с показанным образцом ситуациях);

- метод проблемного изложения;

- эвристический (метод обучения заключается в организации активного поиска решения выдвинутых в обучении (или самостоятельно сформулированных) познавательных задач в ходе подготовки и реализации творческих проектов);

- исследовательский.

Педагогические технологии: информационные технологии, проектная технология, здоровьесберегающие технологии, технология проблемного обучения.

В процессе обучения учащиеся работают с тренажерами быстрого набора текста, средой разработки Code::Blocks и PyCharm, автоматизированными системами проверки решений.

Здоровьесберегающие технологии позволяют создать максимально возможные условия для сохранения, укрепления и развития эмоционального,

интеллектуального и физического здоровья, в том числе в условиях работы с компьютерной техникой.

Проблемное обучение — это тип развивающего обучения, содержание которого представлено системой проблемных задач различного уровня сложности, в процессе решения которых учащиеся овладевают новыми знаниями и способами действия, а через это происходит формирование творческих способностей: продуктивного мышления и познавательной мотивации.

### **Учебно-методические средства обучения**

Для реализации программы используется:

- специализированная литература по направлению, подборка журналов,
- образцы программ и систем, выполненные учащимися и педагогом,
- учебно-методические пособия для педагога и учащихся, включающие дидактический, информационный, справочный материалы на различных носителях, компьютерное и видео оборудование.

Применяемое на занятиях дидактическое и учебно-методическое обеспечение включает в себя электронные учебники, справочные материалы и системы используемых Программ, Интернет.

### **Кадровое обеспечение**

Программу реализуют педагоги структурного подразделения Центр цифрового образования детей «IT-куб».

### **Информационное обеспечение**

Для реализации программы планируется использование следующих информационных ресурсов:

- курс «Введение в программирование C++» (<https://stepik.org/course/363/syllabus>);

- курс «Быстрый старт в спортивное программирование»  
(<https://stepik.org/course/64454/syllabus>);
- курс «Основы C/C++ для спортивного программирования»  
(<https://stepik.org/course/80538/syllabus>);
- курс «Основы Теории Графов для спортивного программирования»  
(<https://stepik.org/course/111246/syllabus>).

## СПИСОК ЛИТЕРАТУРЫ

### Список литературы для педагогов

1. Босова Л. Л. Информатика. 8 класс : учебник. / Босова Л. Л. — М.: БИНОМ. Лаборатория знаний, 2016. — 176 с.
2. Бхаргава А. Грокаем алгоритмы: иллюстрированное пособие для программистов и любопытствующих/ А. Бхаргава. — СПб.: Питер, 2017. — 288 с.
3. Луридас П. Алгоритмы для начинающих: теория и практика для разработчика. / П. Луридас. — М.: Эксмо, 2018. — 608 с.
4. Первин Ю. А. Методика раннего обучения информатике. / Первин Ю. А. — М.: БИНОМ. Лаборатория знаний, 2008. — 228 с.
5. Поляков К. Ю. Информатика. 7 класс (в 2 частях): учебник. Ч. 1 / Поляков К. Ю., Еремин Е. А. — М.: БИНОМ. Лаборатория знаний, 2019. — 160 с.
6. Семакин, И. Г. Информатика и ИКТ: учебник для 9 класса. / Семакин, И. Г., Залогова, Л. А. и др. М: БИНОМ. Лаборатория знаний, 2014. — 171 с.

### Список литературы для учащихся

1. Кэрл Вордерман, Джон Вудкок и Шон Макаманус «Программирование для детей». - Первое издание изд. - Манн, Иванов и Фербер, 2015. - 224 с.
2. Поляков К. Ю. Информатика. 7 класс (в 2 частях): учебник. Ч. 1 / Поляков К. Ю., Еремин Е. А. — М.: БИНОМ. Лаборатория знаний, 2019. — 160 с.
3. Семакин, И. Г. Информатика и ИКТ: учебник для 9 класса. / Семакин, И. Г., Залогова, Л. А. и др. М: БИНОМ. Лаборатория знаний, 2014. — 171 с.

### Примерные задания для промежуточной аттестации

Каждая задача оценивается в 20 баллов.

1. Даны две сцепленные шестеренки. У одной шестеренки  $N$  зубцов, у другой –  $K$ . Требуется найти, какое минимальное число поворотов на один зубчик требуется сделать, чтобы шестеренки вернулись в исходное состояние.

Входные данные

В единственной строке два натуральных числа  $N$  и  $K$ , не превосходящих 10 миллионов.

Выходные данные

Выведите искомое количество зубчиков. Гарантируется, что оно не более миллиарда.

2. Длина Московской кольцевой автомобильной дороги — 109 километров. Байкер Вася стартует с нулевого километра МКАД и едет со скоростью  $v$  километров в час. На какой отметке он остановится через  $t$  часов?

Входные данные

Программа получает на вход значения  $v$  и  $t$ . Если  $v > 0$ , то Вася движется в положительном направлении по МКАД, если же значение  $v < 0$ , то в отрицательном.

Выходные данные

Программа должна вывести целое число от 0 до 108 — номер отметки, на которой остановится Вася.

3. А и В готовятся к олимпиадам по программированию.

Чтобы развить свое логическое мышление и лучше решать задачи, А и В решили сыграть в шахматы. Во время игры А задумался, чья же позиция в данный момент сильнее.

Для каждой шахматной фигуры известен ее вес:

вес ферзя равен 9,

вес ладьи равен 5,

вес слона равен 3,

вес коня равен 3,

вес пешки равен 1,

вес короля не учитывается при оценке позиции.

Вес позиции игрока равен сумме весов всех его фигур на доске.

Так как А не любит считать, он попросил вас помочь определить, вес позиции какого из двух игроков больше.

Входные данные

На вход подаются восемь строк, по восемь символов каждая — описание доски.

На доске белые фигуры обозначаются заглавными буквами, черные фигуры — строчными.

Белые фигуры обозначаются следующим образом: ферзь обозначается символом 'Q', ладья — 'R', слон — 'B', конь — 'N', пешка — 'P', король — 'K'.

Черные фигуры обозначаются, соответственно, 'q', 'r', 'b', 'n', 'p', 'k'.

Пустая клетка доски обозначается символом '.' (точка).

Не гарантируется, что заданная шахматная позиция достижима во время реальной игры. В частности, каждой фигуре может быть произвольное (в том числе нулевое) количество, король может находиться под боем, и так далее.

Выходные данные

Выведите "White" (без кавычек), если вес позиции белых больше веса позиции черных, "Black" если вес позиции черных больше веса позиции белых, и "Draw" если веса позиций белых и черных равны.

Примеры

входные данные

...QK...

.....

.....

.....

.....

.....

.....



...rk...

выходные данные

White

входные данные

rnbqkbnr

pppppppp

.....

.....

.....

.....

PPPPPPPP

RNBQKBNR

выходные данные

Draw

входные данные

grrrrrrr

...k....

.....

.....

.....

.....

K...Q...

.....

выходные данные

Black

Примечание

В первом тестовом примере вес позиции белых равен 9, вес позиции черных равен 5.

Во втором тестовом примере веса позиций черных и белых равны 39.

В третьем тестовом примере вес позиции белых равен 9, вес позиции черных равен 16.

4. Вводится сначала число  $N$ , а затем  $N$  чисел. Выведите эти  $N$  чисел в обратном порядке.

Входные данные

Вводится число  $N$  ( $0 < N < 100$ ), а затем  $N$  чисел из диапазона Integer.

Выходные данные

Выведите  $N$  чисел в обратном порядке

Пример входных данных

7

2 4 1 3 5 3 1

Пример выходных данных

1 3 5 3 1 4 2

5. В квадратной таблице  $N \times N$  подсчитать суммы чисел, стоящих на диагоналях.

Входные данные

Во входном файле содержится число  $N$  ( $1 \leq N \leq 100$ ), а затем матрица  $N \times N$ .

Элементы матрицы - числа из диапазона integer.

Выходные данные

В выходной файл выдать сначала сумму чисел на главной,

а затем - на побочной диагонали.

Пример входных данных

3

1 2 3

4 5 6

10 9 8

Пример выходных данных

14 18

## Приложение 2

### Примерные задания для итоговой аттестации

Задачи на программирование. Максимальный балл за решение каждой задачи 20 баллов.

1. Полоска бумаги имеет размеры  $A \times B$ . Каждый раз от нее отрезается квадрат максимального размера до тех пор, пока не получится квадрат. Сколько квадратов получится?

Входные данные

Программе даны числа  $A$  и  $B$  ( $1 \leq A, B \leq 10^9$ ).

Выходные данные

Требуется вывести количество квадратов.

2. В связи с визитом Императора Палпатина было решено обновить состав дроидов в ангаре 32. Из-за кризиса было решено новых дроидов не закупать, но выкинуть пару старых. Как известно, Палпатин не переносит дроидов с маленькими серийными номерами, так что все, что требуется - найти среди них двух, у которых серийные номера наименьшие.

Входные данные

Первая строка входного файла содержит целое число  $N$  – количество дроидов. ( $2 \leq N \leq 1000$ ), вторая строка –  $N$  целых чисел, по модулю не превышающих  $2 \cdot 10^9$  – номера дроидов

Выходные данные

Выведите два числа: первым – последний по величине из номеров дроидов (такого следует утилизировать в первую очередь), а вторым – предпоследний.

3. У меня в прихожей стоят в ряд 20 тапочек – 10 левых и 10 правых. Приходя домой, я переобуваюсь и выбираю два тапочка – левый и правый, в которые мне удобнее всего засунуть ноги. Естественно, что левый тапочек должен стоять левее правого, и расстояние (количество других тапочек) между ними должно быть как можно меньше. Напишите программу, которая вычисляет, сколько же тапочек стоит между теми, которые мне удобнее всего надеть.

Входные данные

Вводится последовательность из 10 нулей и 10 единиц, записанных в некотором порядке. Единица соответствует левому тапочку, 0 – правому тапочку. Числа разделены пробелами.

Выходные данные

Программа должна вывести количество тапочек между самыми удобными тапочками, или -1, если таких нет.

4. Вдоль границы двух государств когда-то была построена новая стена. Она была собрана из одинаковых кубических блоков и ее высота по всей длине была одинаковой и равнялась 5 блокам. Много лет этого было достаточно, чтобы удержать соседние королевства от нападения друг на друга. Однако инспекция, посланная одним из королей к стене, обнаружила, что во многих вертикальных рядах один или несколько верхних блоков разрушились или упали.

Инспекция составила отчет, в котором для каждого вертикального ряда блоков указана его нынешняя высота. Военное министерство сразу же заинтересовалось вопросом: где находится самый уязвимый участок стены? Участок стены является уязвимым, если он целиком состоит из подряд идущих рядов, высота которых меньше 5 и ограничен с обеих сторон либо границами стены, либо рядами блоков максимальной высоты.

Например, у стены на рисунке два уязвимых участка (второй и третий ряд; пятый и шестой ряды, считая слева). Длина стены на рисунке равна 6.

Один участок стены считается более уязвимым чем другой, если для его полного восстановления понадобится больше блоков.

Для стены на рисунке второй участок более уязвимый чем первый.

Определите номера рядов, с которого начинается и которым заканчивается самый уязвимый участок стены, а также количество блоков, которые необходимы для его полного восстановления. Если возможны несколько вариантов ответа, выведите любой.

Входные данные

Сначала вводится число  $N$  – длина стены (натуральное, не превышает 1000), затем следует  $N$  целых чисел в диапазоне от 0 до 5 – высота соответствующего вертикального ряда. Гарантируется, что на стене есть хотя бы один уязвимый участок.

Выходные данные

Выведите через пробел номер блока, с которого начинается самый уязвимый участок, затем номер, которым он заканчивается, затем количество блоков, которые нужны для полного восстановления этого участка.

4. Дан массив целых чисел. Найти отрезок этого массива с максимальной суммой.

Входные данные

В первой строке дано натуральное число  $n$  ( $1 \leq n \leq 10^5$ ) — размер массива. Во второй строке через пробел перечислены элементы массива. Числа не превышают  $10^4$ .

Выходные данные

Выведите три числа — индекс начала отрезка, индекс конца и самую максимальную сумму. Массив индексируется с единицы. Если ответов несколько — выведите любой.

5. — О, милая сердцу Бобрунья, не желаете ли вы прогуляться со мной по расчудесной лесополосе?

— Конечно, мой Умный Бобр, давайте насладимся прекрасными видами вместе. Как насчет вечера пятницы?

Умный Бобер засуетился. К пятнице все должно быть идеально, поэтому срочно нужно подготовить лесополосу к предстоящей прогулке — выпилить некоторые деревья.

Рассмотрим лесополосу как последовательность деревьев. Каждое дерево  $i$  характеризуется своей эстетической привлекательностью  $a_i$  — одни деревья очень красивы, другие так себе, а третьи даже пугают своим внешним видом!

Умный Бобер вычислил, что для завоевания сердца Бобруньи нужно добиться следующих эффектов:

- во-первых, Бобрунью нужно порадовать: сумма эстетических привлекательностей оставшихся в лесополосе деревьев должна быть максимально возможной;
- во-вторых, Бобрунью нужно удивить: эстетические привлекательности первого и последнего деревьев лесополосы должны быть равны;
- ну и, конечно, прогулка должна состояться: нужно оставить хотя бы два дерева в лесополосе.

Теперь помогите Умному Бобру! Какие деревья придется выпилить для завоевания сердца Бобруньи?

#### Входные данные

В первой строке содержится единственное целое число  $n$  — изначальное количество деревьев в лесополосе,  $2 \leq n$ . Во второй строке через пробел перечислены целые числа  $a_i$  — эстетические привлекательности каждого из деревьев. Все эстетические привлекательности не превосходят  $10^9$  по модулю.

#### Выходные данные

В первой строке выведите два целых числа — суммарную эстетическую привлекательность лесополосы после ее обработки Умным Бобром и количество выпиливаемых деревьев  $k$ .

В следующей строке выведите  $k$  чисел — номера деревьев, которые стоит выпилить. Считайте, что деревья пронумерованы от 1 до  $n$  слева направо.

Если существует несколько решений, то выведите любое из них. Гарантируется, что хотя бы два дерева имеют одинаковую эстетическую привлекательность.

### Примеры

входные данные

5

1 2 3 1 2

выходные данные

8 1

1

входные данные

5

1 -2 3 1 -2

выходные данные

5 2

2 5