Краевое государственное автономное
профессиональное образовательное учреждение
«Кунгурский колледж агротехнологий и управления»

# МЕТОДИЧЕСКИЕ УКАЗАНИЯ
# ПО ВЫПОЛНЕНИЮ ПРАКТИЧЕСКИХ И ЛАБОРАТОРНЫХ РАБОТ
# ПО ДИСЦИПЛИНЕ
# ОП.04 ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ
## 09.02.07 «Информационные системы и программирование»

2023 г.

Рассмотрено на заседании МК информационных дисциплин от «30» августа 2023 г.

Председатель МК
_____А.В.Атушкина

Утверждаю
Зам. директора
_____Л.И.Петрова

Организация-разработчик: ГБПОУ «ККАТУ»

Составитель: А.В.Атушкина

## Пояснительная записка

Методические указания по выполнению лабораторных и практических работ по ОП 04. «Основы алгоритмизации и программирования» разработаны в соответствии с рабочей программой дисциплины и предназначены для приобретения необходимых практических навыков и закрепления теоретических знаний, полученных обучающимися при изучении профессионального модуля, обобщения и систематизации знаний перед экзаменом.

Методические указания предназначены для обучающихся специальности 09.02.07 Информационные системы и программирование.

Дисциплина ОП 04. «Основы алгоритмизации и программирования» относится к общепрофессиональному циклу, изучается на 2 курсе и при его изучении отводится значительное место выполнению практических работ.

Освоение содержания ОП 04. «Основы алгоритмизации и программирования» во время выполнения практических работ обеспечивает достижение обучающимися следующих **результатов:**

| Код ОК | Наименование |
|---|---|
| ОК 1. | Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам |
| ОК 2. | Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности |
| ОК 4. | Эффективно взаимодействовать и работать в коллективе и команде |
| ОК 5. | Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста |
| ОК 9. | Пользоваться профессиональной документацией на государственном и иностранном языках |

| ***Код*** | ***Наименование видов деятельности и профессиональных компетенций*** |
|---|---|
| ***ВД 2*** | Осуществление интеграции программных модулей |
| ПК 2.4. | Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения |
| ПК 2.5. | Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования |

| Код трудовой функции | Наименование |
|---|---|
| ТФ A/04.4 | **Трудовые действия**<br>– Проведение тестирования разрабатываемого модуля ИС в соответствии с трудовым заданием;<br>– Устранение обнаруженных несоответствий;<br>– Фиксирование результатов тестирования в системе учета. |
| | **Необходимые умения**<br>– Кодировать на языках программирования;<br>– Тестировать результаты собственной работы. |
| | **Необходимые знания**<br>– Языки программирования и работы с базами данных;<br>– Основы современных операционных систем;<br>– Основы современных систем управления базами данных;<br>– Устройство и функционирование современных ИС;<br>– Теория баз данных;<br>– Системы хранения и анализа баз данных;<br>– Современные методики тестирования разрабатываемых ИС. Инструменты и методы модульного тестирования;<br>– Источники информации, необходимой для профессиональной деятельности;<br>– Современный отечественный и зарубежный опыт в профессиональной деятельности;<br>– Культура речи;<br>– Правила деловой переписки. |
| ТФ A/12.4 | **Трудовые действия**<br>– Проведение физического аудита в области качества в соответствии с трудовым заданием;<br>– Инициирование коррекции (запросов на устранение обнаруженных несоответствий) по результатам аудитов. |
| | **Необходимые умения**<br>– Работать с записями по качеству (в том числе с корректирующими действиями, предупреждающими действиями, запросами на исправление несоответствий). |

| | |
|---|---|
| | **Необходимые знания**<br>– Инструменты и методы проведения физических аудитов качества;<br>– Основы современных операционных систем;<br>– Источники информации, необходимой для профессиональной деятельности;<br>– Современный отечественный и зарубежный опыт в профессиональной деятельности;<br>– Культура речи;<br>– Правила деловой переписки. |

Рабочая программа дисциплины предусматривает проведение практических работ в объеме 80 часов.

**Порядок выполнения практической работы**

- записать название работы, ее цель в тетрадь;
- выполнить основные задания в соответствии с ходом работы;
- выполнить индивидуальные задания.

**Рекомендации по оформлению практической работы**

Задания выполняются обучающимися по шагам. Необходимо строго придерживаться порядка действий, описанного в практической работе.

Результаты выполнения практических работ необходимо сохранять в своей папке на компьютере или USB – накопителе.

В случае пропуска занятий обучающийся осваивает материал самостоятельно в свободное от занятий время и сдает практическую работу с пояснениями о выполнении.

## Требования к технике безопасности при выполнении лабораторных/практических работ

Вводный инструктаж

Инструкция №1 ПРАВИЛА ПОВЕДЕНИЯ СТУДЕНТОВ В КАБИНЕТЕ ИНФОРМАТИКИ

Кабинет информатики относится к кабинетам повышенной опасности, находясь в нем необходимо соблюдать требования настоящей инструкции.

1. Не заходите в кабинет без разрешения преподавателя.

2. Во время перемены все студенты выходят в коридор. В кабинете кроме преподавателя могут находиться только дежурные.

3. Запрещается находиться в кабинете в верхней одежде, грязной обуви.

4. Не бегайте по кабинет - можете получить травму или повредить оборудование.

5. Не бросайте мусор в кабинете, этим вы создаете пожарную опасность.

6. Будьте внимательны и дисциплинированны, точно выполняйте указания преподавателя.

7. Не приступайте к выполнению работы без разрешения преподавателя.

8. Не пользуйтесь электрическими розетками для шалости, это опасно для вашей жизни.

Инструктаж №2 ПРАВИЛА РАБОТЫ СТУДЕНТОВ ЗА КОМПЬЮТЕРОМ ПЕРЕД ВЫПОЛНЕНИЕМ РАБОТЫ необходимо выслушать преподавателя о ее содержании и ходе выполнения.

ВО ВРЕМЯ РАБОТЫ

1. Компьютер можно включать только после разрешения преподавателя.

2. ЗАПРЕЩАЕТСЯ:

-прикасаться к проводам, лишенных изоляции;

-включать компьютер со снятым корпусом;

-производить подключение устройств к включенному компьютеру;

-прикасаться руками и острыми предметами к экрану монитора, внутренним частям компьютера;

-есть за компьютером;

-включать компьютер сразу же после его выключения (необходимо подождать 10-15 секунд).

3. Обнаружив неисправность в электрических устройствах, находящихся под напряжением, необходимо немедленно отключить источник электропитания и сообщить об этом преподавателю.

4. Не оставляйте рабочего места без разрешения преподавателя.

ПОСЛЕ ОКОНЧАНИЯ РАБОТЫ

1. Корректно завершите работу компьютера.

2. Сдай рабочее место преподавателю.

Инструктаж No3 ПРАВИЛА ПОЖАРНОЙ БЕЗОПАСНОСТИ

1. В кабинете должны, быть средства для тушения пожара: огнетушитель.

2. Кабинет должен содержаться в чистоте. Весь сгораемый мусор следует систематически выносить.

3. В кабинете не допускается хранение легковоспламеняющихся жидкостей.

4. ЗАПРЕЩАЕТСЯ:

-допускать к работе студентов, не ознакомленных с правилами техники безопасности;

-оставлять без присмотра включенное в электрическую сеть оборудование;

-подвешивать электропроводку на гвоздях, использовать электропровода с поврежденной изоляцией, некалиброванные предохранители, обертывать электрические лампы бумагой или тканью, подвешивать стенды, таблицы и пр. на электропровода;

-работа в кабинете с нагревательными приборами;

-производить сложный ремонт компьютерной техники.

5. По окончании работы необходимо тщательно осмотреть помещение, устранить все недочеты, отключить напряжение электросети с помощью рубильника.

6. В случае возникновения пожара необходимо:

-отключить напряжение электрической сети;

-немедленно эвакуировать студентов;

-принять меры по тушению пожара;

-сообщить о пожаре по телефону 01 или 112.

## Перечень практических работ

1. Пр.р.№1. Составление блок-схем алгоритмов
2. Пр.р.№2. Ввод и отладка простейших линейных программ
3. Пр.р.№3. Целочисленная арифметика. Задачи на целочисленное деление
4. Пр.р.№4. Задачи на построение и расчет математических выражений.
5. Пр.р.№5. Логические операции и выражения
6. Пр.р.№6. Составление программ с процедурами ввода с клавиатуры и вывода на дисплей
7. Пр.р.№7. Составление программ с использованием ветвления
8. Пр.р.№8. Составление программ с использованием цикла FOR
9. Пр.р.№9. Составление программ с использованием цикла While

10. Пр.р.№10. Составление программ с использованием цикла Repeat
11. Пр.р.№11. Работа с элементами массива
12. Пр.р.№12. Сортировка массива
13. Пр.р.№13. Работа с двумерными массивами
14. Пр.р.№14. Составление и отладка программ с использованием процедур
15. Пр.р.№15. Составление и отладка программ с использованием функции
16. Пр.р.№16. Составление задач рекурсивного типа
17. Пр.р.№17. Работа с графикой
18. Пр.р.№18. Построение простых графических объектов
19. Пр.р.№19. Ввод и отладка программы по обработке строк
20. Пр.р.№20. Ввод и отладка программы с использованием множеств
21. Пр.р.№21. Работа с файлом последовательного доступа
22. Пр.р.№22. Работа с файлом произвольного доступа
23. Пр.р.№23. Программирование модуля
24. Пр.р.№24. Создание библиотеки подпрограмм
25. Пр.р.№25. Создание консольного приложения
26. Пр.р.№26. Создание текстового редактора
27. Пр.р.№27. Изучение интегрированной среды разработчика
28. Пр.р.№28. Рисование мышью на канве
29. Пр.р.№29. Создание проекта с использованием различных компонентов
30. Пр.р.№30. Разработка оконного приложения
31. Пр.р.№31. Разработка многооконного приложения
32. Пр.р.№32. Изучение главного меню среды. Помещение объектов на форму и задание им свойств
33. Пр.р.№33. Разработка оконного приложения
34. Пр.р.№34. Создание класса. Создание проекта с использованием созданного класса
35. Пр.р.№35. Оформление кода программы в соответствии со стандартом кодирования

## Критерии оценки выполнения практических работ

| Оценки | Критерии оценок |
|---|---|
| «5» | - обучающийся подбирает необходимые для выполнения предлагаемых работ источники знаний (литература, материалы, инструменты), показывает необходимые для проведения практической работы теоретические знания. Правильно оформлена практическая работа, соблюдена технологическая последовательность выполнения данного вида работ. Работа оформлена аккуратно. |
| «4» | - практическая работа выполняется обучающимся в полном объёме и самостоятельно. Обучающийся использует указанные преподавателем источники информации. Могут быть неточности и небрежность в оформлении работы. Работа показывает знания обучающимися основного теоретического материала, но имеются незначительные ошибки при оформлении практической части роботы. |
| «3» | - обучающийся выполняет и оформляет практическую работу полностью с помощью преподавателя или хорошо подготовленных и уже выполнивших на «отлично» данную работу других обучающихся. |
| «2» | - практическая работа не выполнена полностью за отведенное время по неуважительной причине. |

## Перечень учебных изданий, Интернет-ресурсов, дополнительной литературы

Основные источники:

1. Семакин И.Г., Шестаков А.П Основы алгоритмизации и программирования. – М.: Издательский центр «Академия», 2021. – 304 с.

Основные электронные издания:

1. Трофимов В. В. Основы алгоритмизации и программирования: учебник для среднего профессионального образования / В. В. Трофимов, Т. А. Павловская; под редакцией В. В. Трофимова. – М.: Издательство Юрайт, 2021. – 137 с. – (Профессиональное образование). – ISBN 978-5-534-07321-8. – Текст: электронный // Образовательная платформа Юрайт [сайт]. – URL: https://urait.ru/bcode/473347 (дата обращения: 20.08.2023).

Дополнительные источники:

1. Кудрина Е.В. Основы алгоритмизации и программирования на языке C#: учебное пособие для среднего профессионального образования / Е. В. Кудрина, М. В. Огнева. – М.: Издательство Юрайт, 2021. – 322 с. – (Профессиональное образование). – ISBN 978-5-534-10772-2. – Текст: электронный // Образовательная платформа Юрайт [сайт]. – URL: https://urait.ru/bcode/475228 (дата обращения: 20.08.2023).

***Практическая работа.№7***

***Составление программ с использованием ветвления***

***Цель работы:*** *Познакомиться с оператором условного перехода, составным и пустым оператором.*

***Основные понятия***

***Составной оператор и пустой оператор.*** *Составной оператор – это последовательность произвольных операторов программы, заключенная в операторные скобки – зарезервированные слова BEGIN … END. Составные операторы – важный инструмент Pascal, дающий возможность писать программы по современной технологии структурного программирования (без операторов перехода GOTO).*

*Pascal допускает произвольную глубину вложенности составных операторов:*

```
BEGIN
……
        BEGIN
        ……
                BEGIN
                ……
                END;
        ……
        END;
……
END.
```

*Поскольку BEGIN и END представляют собой структурные скобки, то после BEGIN и перед END ставить знак «;» не обязательно.*

*В программе может применяться пустой оператор, не выполняющий никакого действия.*

***Метка и оператор перехода.*** *Метка – это произвольный идентификатор, позволяющий именовать некоторый оператор программы и таким образом ссылаться на него. В Паскале метка – это целое без знака. Метка располагается непосредственно перед помеченным оператором и отделяется от него двоеточием. Описание меток состоит из зарезервированного слова* Label.

*Оператор перехода указывает, что дальнейшая работа должна продолжится в другой части текста программы, а именно с того места, где находилась метка.*

*Структура оператора перехода имеет вид:*

***GOTO*** *<метка>,*

*где GOTO – зарезервированное слово(«перейти на метку») <метка>- произвольный идентификатор.*

***Оператор условного перехода.*** *Оператор условного перехода (условный оператор) позволяет проверить некоторое условие и в зависимости от результатов проверки выполнить то или иное действие.*

*Таким образом условный оператор – это средство ветвления вычислительного процесса.*

*Структура условного оператора имеет вид:*

***IF*** *условие* ***THEN*** *оператор1* ***ELSE*** *оператор2*,

*где IF, THEN, ELSE – зарезервированные слова («если», «то», «иначе»); оператор1, оператор2 – любые операторы языка Паскаль (в том числе и составные).*

*Условный оператор работает по следующему алгоритму. Вначале вычисляется условие, если результат* True *(истина), то выполняется оператор1, а оператор2 пропускается; если результат* False *(ложь), то, наоборот, оператор1 пропускается, а выполняется оператор2.*

*Оператор IF может быть неполным, т.е. часть «ELSE оператор2» может быть опущена. Тогда при значении* True *условного выражения выполняется оператор1, в противном случае он пропускается.*

*Если оператор1 и оператор2 – составные, то условный оператор будет иметь вид:*

***IF условие THEN***

***BEGIN***

***……***

***END***

***ELSE***

***BEGIN***

***……***

***END.***

***Пример.*** *Напишите программу, определяющую наименьшее значение из двух чисел, введенных с клавиатуры.*

*Решение*

```
PROGRAM Minimum;
VAR a, b, min : real;
BEGIN
WRITELN('Введите два числа');
READLN(a, b);
IF a<b Then min:=a ELSE min:=b;
WRITELN(min);
END.
```

**Вариант 1**

*1.Написать программу, которая запрашивает у пользователя номер одного из весенних месяцев, и выводит количество дней в этом месяце. Программа должна проверять, является ли введенный месяц весенним.*

*2.Написать программу, которая классифицирует компьютерную сеть. Программа запрашивает у пользователя число компьютеров в сети и в зависимости от введенного количества выводит класс сети (если число ЭВМ меньше 256– то это сеть класса С, от 256 до 65535 – сеть класса В, свыше 65535 – сеть класса А).*

*3.Написать программу, которая вычисляет частное от деления двух чисел. Программа должна проверять правильность введенных пользователем данных и, если они неверные (делитель равен нулю), выдавать сообщение об ошибке.*

*4.Используя, составной оператор, написать программу, которая вычисляет заданное смешанное арифметическое выражение для данных в форматах REAL (переменные a,b) и INTEGER (остальные переменные):*

$$\frac{2*c - d + \sqrt{23}}{\frac{a}{4} - 1}$$

*5.Используя, оператор перехода и метку, написать программу, которая вычисляет*

*заданное целочисленное выражение для данных a, b в формате INTEGER, результат X - тоже целочисленный (INTEGER или LONGINT):*

$$X = \begin{cases} a * b + 1, & \text{если a > b,} \\ 25, & \text{если a = b,} \\ (a - 5) / b, & \text{если a < b;} \end{cases}$$

**Вариант 2**

*1.Написать программу, которая запрашивает у пользователя номер одного из осенних месяцев, и выводит количество дней в этом месяце. Программа должна проверять, является ли введенный месяц осенним.*

*2.Написать программу, которая выводит на экран приглашение: «Который час?», вводит с клавиатуры число X, имеющее смысл времени суток, и печатает слова «Доброе утро», «Добрый вечер», «Добрый день» в зависимости от введенного времени. Программа должна реагировать на ввод неправильного времени: меньше 0 или больше 24.*

*3.Написать программу вычисления сопротивления электрической цепи, состоящей из двух сопротивлений. Сопротивления могут быть соединены последовательно или параллельно.*

*4.Используя, составной оператор, написать программу, которая вычисляет заданное смешанное арифметическое выражение для данных в форматах REAL (переменные a,b) и INTEGER (остальные переменные):*

$$\frac{-2*c - \ln(d) + 53}{\frac{a}{4} - 1}$$

*5.Используя, оператор перехода и метку, написать программу, которая вычисляет заданное целочисленное выражение для данных a, b в формате INTEGER, результат X - тоже целочисленный (INTEGER или LONGINT):*

$$X = \begin{cases} a / b + 1, & \text{если a < b,} \\ -1, & \text{если a = b,} \\ (a * b - 5) / a, & \text{если a > b;} \end{cases}$$

***Практическая работа №19***

***Ввод и отладка программы по обработке строк***

***Цель работы:*** *Научится работать со строками.*

***Основные понятия***

***Операции над строками***

*Для работы с символьной информацией в TP используют новый тип данных - строковый, именуемый ключевым словом STRING (или просто строка). Этот тип данных во многом похож на одномерный массив символов (Array[0..N] of char), но длина строки (максимальное количество символов N ограничивается числом 255). Значение N определяется при объявлении типа STRING(N) и может быть любой константой порядкового типа, но не больше 255. Значение N при объявлении типа STRING можно не указывать: в этом случае длина строки принимается равной 255 символам.*

*Строка в TP трактуется как цепочка символов и к любому символу в строке можно обратиться по адресу (индексу), подобно одномерным массивам типа Array[0..N] of char. Самый первый байт в строке, имеющий адрес 0 (ноль), содержит код, равный числу символов в строке (длине строки).*

*Например, дана строка, имеющая следующее описание:*

*Var St:string;*

*Тогда длину строки St можно определить как значение функции Ord(St[0]).*

*Значением строки может быть любая последовательность символов, заключенная в одинарные кавычки (апострофы). Можно присваивать строке пустое значение, обозначаемое как '' (две одинарные кавычки подряд). При попытке записать в переменную строку длиннее, чем задано в описании, "лишняя" часть будет отсечена.*

*Строки можно присваивать, сливать и сравнивать.*

*Например:*

```
Var st1,st2,st3,sts:string;
  Begin
          . . .
          { Операции присваивания}
  st1:='Фамилия';
  st2:='Имя';
  st3:='Отчество';
          { Операция слияния}
  sto:=st1+' '+st2+' '+st3;
  { В результате в строке sto будет 'Фамилия Имя Отчество'}
  End;              . . .
```

*Сравнение строк основывается на порядке расположения символов в таблице ASCII. Например:*

*'abcd' = 'abcd' - результат сравнения True (истина);*

*'abc'<'bcd' так как код символа 'a' меньше кода символа 'b' и т.п.*

*Для работе со строками в ТР разработан ряд стандартных процедур и функций.*

*Первоначально любая описанная в разделе Var строка содержит "мусор" и рекомендуется инициализировать (заполнять) строки пустыми значениями или чем-либо другим. Для заполнения достаточно длинных строк одинаковыми символами используется встроенная процедура FillChar, например:*

```
Var S:string[80];
Begin    . . .
  FillChar(S[1],80,' '); {Заполнение строки пробелами}
  S[0]:=Chr(80);  {Занесение кода длины строки}
          . . .
End;
```

*Длину строки можно определить также, используя встроенную функцию Length(S), где S - строка типа String.*

*В ряде случаев возникает необходимость преобразования числовых значений в строку и наоборот. Для этого можно использовать две процедуры:*

*1) STR(X,S) - преобразует числовое значение X в строковое S. Возможно задание формата для X в виде: X:F:n (для вещественных чисел, где F - общее число позиций выделяемых под число, включая десятичную точку, а n - число знаков в дробной части) или X:F (для целых чисел). Эта функция чаще всего используется при работе с процедурами модуля GRAPH. Например:*

*STR(55,s); - строковая переменная s принимает значение, равное '55'.*

*2) VAL(S,X,ErrCode) - преобразует строку S в числовое значение, если это возможно. Параметр ErrCode содержит ноль, если преобразование прошло успешно, и тогда в X помещается результат преобразования, в противном случае он содержит номер позиции в строке S, где обнаружен ошибочный символ. Например:*

*Val('125',K,kod) - в результате выполнения этой процедуры переменная K получает целое значение, равное 125, параметр kod=0;*

*Val(' 1.05',M,code) - M=1.05, code=0;*

*Val('100, ',N,code) - это ошибочный вызов, т.к. в исходной строке на 4-й позиции располагается недопустимый для числа символ ',' и поэтому параметр code=4, а переменная N остается без изменения.*

*Кроме перечисленных, в ТР имеется еще 5 функций и процедур:*

*Функция Copy*

*Функция* ***Copy*** *позволяет копировать фрагмент некоторой строки из одной переменной в другую. Вызывая функцию copy, необходимо указывать следующие параметры:*

1) *имя строки, из которой должен извлекаться копируемый фрагмент;*
2) *позицию в строке, начиная с которой будет копироваться фрагмент;*
3) *число копируемых символов.*

*Пример.*

```
var
ws: string [79];
w1,w2,w3 : string [20];
begin
ws := 'фотографирование';
w1 := copy (ws, 1, 4); writeln (w1);
w2 := copy (ws, 5, 4); writeln (w2);
w3 := copy (ws, 10, 3); writeln (w3);
end.
```

*В результате выполнения программы на экран выводится:*

*фото*

*граф*

*РОВ*

*Сообщения об ошибке не будет в случаях, если начальная или конечная позиции копируемого фрагмента находятся вне пределов исходной строки символов. Результатом выполнения операции в первом случае будет строка нулевой длины, во втором - фрагмент от начальной позиции копирования до конца исходной строки.*

*Функция Pos*

*С помощью функции* **Pos** *осуществляется поиск определенного фрагмента в строке. Если заданный фрагмент в строке присутствует, то функция возвращает номер позиции в строке, с которой фрагмент начинается. Если в строке фрагмент не найден, то функция возвращает ноль.*

```
var
ws string [79];
sw string [20];
p : byte;
begin
```

```
ws := 'Электрификация';
sw := 'Эл'; p := Pos (sw,ws); writeln(p);
sw := 'три'; p := Pos (sw,ws); writeln(p);
sw := 'к'; p := Pos (sw,ws); writeln(p);
end.
```

*В результате выполнения программы на экране появляется:*

*1*

*5*

*4*

*Функция Pos требует полного совпадения искомого фрагмента и фрагмента строки, в которой производится поиск. Большие и маленькие буквы считаются различными символами.*

***Процедура Delete***

*Удаление из строки. Для этого используется процедура Delete (Str,n,t), которая вырезает из строки Str t символов, начиная с n-го, таким образом сама строка изменяется.*

***Пример:*** *Дан фрагмент программы:*

```
Strl:='ABCDEFGH':
Delete(Strl, 3, 4);
Wrlteln(Strl);
```

*После выполнения этих операторов из строки будут удалены четыре символа, начиная с третьего, то есть строка будет такой: Strl='ABGH'.*

***Процедура Insert***

*Вставка подстроки в строку. Это можно сделать, применяя процедуру Insert(Strl,Str2,n) — вставка строки Str1 в строку Str2, начиная с n-го символа, при этом первая строка остается такой же, как и была, а вторая получает новое значение.*

***Пример:***

```
Strl:='ABCDEFGH';
Str2:='abcdefgh';
Insert(Strl, Str2, 3);
```

*В результате выполнения данной процедуры строка будет такой — Str2='abABCDEFGHcdefgh'. Этот же результат будет и после выполнения такой последовательности операторов:*

```
Str2:='abcdefgh';
```

*Insert('ABCDEFGH', Str2, 3);*

### *Вариант 1*

*1. Написать программу, которая запрашивает имя пользователя и здоровается с ним. Рекомендуемый вид экрана во время работы программы приведен ниже (данные, введенные пользователем, выделены полужирным шрифтом).*

*Как Вас зовут?*

*Введите свое имя и нажмите <Enter>*

***-> Вася***

*Здравствуйте, Вася!*

*2. Написать программу, которая определяет, является ли вводимая последовательность символов идентификатором.*

### *Вариант 2*

*1. Написать программу, которая выводит на экран сообщение в "телеграфном" стиле, т. е. буквы сообщения должны появляться по одной, причем с некоторой задержкой.*

*2. Написать программу, которая подсчитывает количество сдвоенных символов «сс», «нн», «лл» во введенном тексте.*

### *Вариант 3*

*1. Написать программу, которая выдает код символа, введенного пользователем. Программа должна завершать работу после ввода, например, точки.*

*2. Написать программу, которая разбивает произвольный текст на строки определенной длины. При переносе слова предусмотреть вывод дефиса.*

### *Вариант 4*

*1. Написать программу, которая выводит на экран первую часть таблицы кодировки символов (символы, имеющие коды от 0до 127). Таблица должна состоять из восьми колонок и шестнадцати строк. В первой колонке должны быть символы, имеющие код от 0 до 15, во второй — От 16 до 31 и т. д.*

*2. Дана символьная строка. Написать программу, которая подсчитывает, сколько раз в ней встречается подслово ABBA.*

### *Вариант 5*

*1. Написать программу, которая в веденной с клавиатуры строке преобразует строчные буквы русского алфавита в прописные (учтите, что функция upCase с символами русского алфавита не работает).*

*2. Написать программу, которая находит во введенном тексте самое длинное и самое короткое слово.*

***Вариант 6***

*1. Написать программу, которая удаляет начальные пробелы из введенной с клавиатуры строки.*

*2. Написать программу, которая из заданной строки исключить все символы, входящие в нее более одного раза.*

***Вариант 7***

*1. Написать программу, которая проверяет, является ли введенная с клавиатуры строка целым числом. Рекомендуемый вид экрана во время работы программы приведен ниже (данные, введенные пользователем, выделены полужирным шрифтом).*

*Введите число и нажмите <Enter> -> **23.5***

*Введенная строка не является целым числом.*

*2. Написать программу, которая проверяет, правильно ли в заданном тексте расставлены круглые скобки.*

***Вариант 8***

*1. Написать программу, которая проверяет, является ли введенная с клавиатуры строка двоичным числом.*

*2. Написать программу, которая в заданной последовательности символов подсчитывает общее количество символов «+», «-», «*» и исключить их из текста.*

***Вариант 9***

*1. Написать программу, которая проверяет, является ли введенная с клавиатуры строка шестнадцатеричным числом.*

*2. Написать программу, в которой вводится последовательность ключевых слов. Отсортировать их по алфавиту.*

***Вариант 10***

*1. Написать программу, которая проверяет, является ли введенная с клавиатуры строка дробным числом.*

*2. Написать программу, которая в заданном тексте переворачивает каждое слово.*

***Вариант 11***

*1. Написать программу, которая преобразует введенное с клавиатуры восьмиразрядное двоичное число в десятичное. Рекомендуемый вид экрана во время работы программы представлен ниже (введенные пользователем данные выделены полужирным шрифтом).*

*Введите восьмиразрядное двоичное число*

*и нажмите <Enter>*

***-> 11101010***

*Двоичному числу 11101010 соответствует десятичное* ***234***

*Для завершения работы программы нажмите <Enter>.*

*2. Написать программу, которая определяет можно ли из символов заданной строки составить вашу фамилию.*

***Вариант 12***

*1. Написать программу, которая преобразует введенное с клавиатуры двухразрядное шестнадцатеричное число в десятичное.*

*2. Написать программу, которая выясняет. верно ли, что среди символов строки произвольной длины имеются все символы, входящие в слово ДЕНЬ.*

***Вариант 13***

*1. Написать программу, которая преобразует введенное пользователем десятичное число в число в указанной системе счисления (от 2-х до. 10-ти). Рекомендуемый вид экрана во время работы программы приведен ниже.*

*Введите целое число —>* ***67***

*Введите основание системы счисления —>* ***2***

*100011*

*Написать программу, которая преобразует введенное пользователем десятичное число в шестнадцатеричное.*

*2. Написать программу, которая из заданной последовательности символов удаляет лишни пробелы, разделяющие слова.*

***Вариант 14***

*1. Написать программу, которая вычисляет значение выражения* $N_0O_1N_1O_2...O_kN_k$, *где* $N_i$ *— целое одноразрядное число, О, — один из двух знаков простейших арифметических действий: сложения или вычитания. Ниже представлен рекомендуемый вид экрана во время работы программы (данные, введенные пользователем, выделены полужирным шрифтом).*

*Введите арифметическое выражение, например, 4+5-3-5+2 и нажмиет <Enter>*

***9-5+4+2-6***

*Значение введенного выражения:* ***4***

*Для завершения программы нажмите <Enter>.*

*2. Написать программу, которая в заданной строке символов исключает все группы символов вида АВС.*

## *Вариант 15*

*1. Составьте программу, меняющую в слове Х все сочетания "ва" на "к".*

*2. Написать программу, которая в приложении, содержащем не менее двух слов. поменять местами первое и последнее слово.*