

Министерство образования и науки Российской Федерации
Федеральное агентство по образованию
Ярославский государственный университет им. П.Г. Демидова
Кафедра математической кибернетики

И н ф о р м а т и к а

Лабораторный практикум

Рекомендовано
Научно-методическим советом университета
для студентов специальности Компьютерная безопасность

Ярославль 2005

УДК 002:372.8
ББК В18я73
И 74

*Рекомендовано
Редакционно-издательским советом университета
в качестве учебного издания. План 2005 года*

Рецензент
кафедра математической кибернетики
Ярославского государственного университета

Составители: О.В. Власова, О.П. Полякова

Информатика: Лабораторный практикум / Сост.:
И 74 О.В. Власова, О.П. Полякова; Яросл. гос. ун-т. – Ярославль:
ЯрГУ, 2005. – 48 с.

Методические указания предназначены для студентов,
обучающихся по специальности 075200 Компьютерная
безопасность (дисциплина «Информатика», блок ОПД),
очной формы обучения.

УДК 002:372.8
ББК В18я73

© Ярославский государственный университет им. П.Г. Демидова, 2005
© О.В. Власова, О.П. Полякова, 2005

I. Информатика

Лабораторная работа № 1

Цель работы. Придумать алгоритм и решить задачу на применение простейших конструкций языка - цикла и условия, осуществляя пошаговый ввод и вывод данных (массив использовать нельзя).

Требуется ввести с клавиатуры натуральное число n , далее в цикле обеспечить ввод и обработку остальных данных.

Варианты.

1. Даны натуральные числа n, a_1, \dots, a_n . Определить количество членов a_i последовательности a_1, \dots, a_n кратных 3 и не кратных 5.

2. Даны натуральные числа n, a_1, \dots, a_n . Определить количество членов a_i последовательности a_1, \dots, a_n , являющихся квадратами четных чисел.

3. Даны натуральные числа n, a_1, \dots, a_n . Определить количество членов a_i последовательности a_1, \dots, a_n , удовлетворяющих условию: $a_i < (a_{i-1} + a_{i+1})/2$

4. Даны натуральные числа n, a_1, \dots, a_n . Определить количество членов a_i последовательности a_1, \dots, a_n , удовлетворяющих условию: $2^i < a_i < i!$

5. Даны натуральные числа n, a_1, \dots, a_n . Определить количество членов a_i последовательности a_1, \dots, a_n , имеющих четные порядковые номера и являющиеся нечетными числами.

6. Даны натуральные числа n, a_1, \dots, a_n . Найти те члены a_i , последовательности a_1, \dots, a_n , которые являются удвоенными нечетными числами.

7. Даны натуральные числа n, a_1, \dots, a_n . Найти те члены a_i , последовательности a_1, \dots, a_n , которые при делении на 7 дают остаток 1, 2 или 5.

8. Даны натуральные числа n, a_1, \dots, a_n . Найти те члены a_i , последовательности a_1, \dots, a_n , которые обладают тем свойством, что корни уравнения $x^2 + 3a_i - 5 = 0$ действительны и положительны.

9. Даны натуральное n , целые числа a_1, \dots, a_n . Получить сумму тех членов последовательности a_1, \dots, a_n , которые кратны 5.

10. Даны натуральное n , целые числа a_1, \dots, a_n . Получить сумму тех членов последовательности a_1, \dots, a_n , которые нечетны и отрицательны.

11. Даны натуральное n , целые числа a_1, \dots, a_n . Получить сумму тех членов последовательности a_1, \dots, a_n , которые удовлетворяют условию $|a_i| < i^2$.

12. Даны натуральное n , целые числа a_1, \dots, a_n . Найти количество и сумму тех членов данной последовательности, которые делятся на 5 и не делятся на 7.

13. Даны натуральные n, p , целые числа a_1, \dots, a_n . Получить произведение тех членов последовательности a_1, \dots, a_n , которые кратны p .

14. Даны натуральное n , действительные числа a_1, \dots, a_n . В последовательности a_1, \dots, a_n определить число соседств двух положительных чисел.

15. Даны натуральное n , действительные числа a_1, \dots, a_n . В последовательности a_1, \dots, a_n определить число соседств двух чисел разного знака.

16. Даны натуральное n , действительные числа a_1, \dots, a_n . В последовательности a_1, \dots, a_n определить число соседств двух чисел одного знака, причем модуль первого должен быть больше модуля второго числа.

17. Даны натуральное n , целые числа a_1, \dots, a_n . Имеются ли в данной последовательности два идущих подряд нулевых члена?

18. Даны натуральное n , целые числа a_1, \dots, a_n . Имеются ли в данной последовательности три идущих подряд нулевых члена?

19. Даны натуральное n , целые числа a_1, \dots, a_n . Получить сумму положительных и число отрицательных членов данной последовательности.

20. Даны натуральное n , целые числа a_1, \dots, a_n . Найти наименьшее из четных чисел, входящих в последовательность $a_1 - 1, a_1, a_2, \dots, a_n$.

21. Даны натуральное n , действительные числа a_1, \dots, a_n . Получить $\max(a_1, \dots, a_n)$.

22. Даны натуральное n , действительные числа a_1, \dots, a_n .
Получить $\max(a_2, a_4, \dots)$.
23. Даны натуральное n , действительные числа a_1, \dots, a_n .
Получить $\min(a_1, a_3, \dots)$.
24. Даны натуральное n , действительные числа a_1, \dots, a_n .
Получить $\min(a_1, \dots, a_n)$.
25. Даны натуральное n , действительные числа a_1, \dots, a_n .
Получить $\max(|a_1|, \dots, |a_n|)$.
26. Даны натуральное n , действительные числа a_1, \dots, a_n .
Верно ли, что отрицательных членов в последовательности a_1, \dots, a_n больше, чем положительных?
27. Даны натуральное n , действительные числа a_1, \dots, a_n .
Верно ли, что наибольший член в последовательности a_1, \dots, a_n по модулю больше единицы?
28. Даны натуральное n , целые числа x, a_1, \dots, a_n . Если в последовательности a_1, \dots, a_n есть хотя бы один член, равный x , то получить сумму всех членов, следующих за ним; в противном случае ответом должно быть число -10.
29. Даны натуральное n , целые числа x, a_1, \dots, a_n .
Определить, каким по счету идет в последовательности a_1, \dots, a_n член, равный x . Если такого члена нет, то ответом должно быть число 0.
30. Даны натуральное n , целые числа a_1, \dots, a_n . Получить последовательность, которая отличается от исходной тем, что все нечетные члены удвоены.

Лабораторная работа № 2

Цель работы. Дана функция, заданная бесконечным рядом. С клавиатуры вводятся x и e (e - точность 0.001 - 0.00001), нужно вычислить бесконечную сумму с заданной точностью e . Вычисления прекращаются, когда очередное слагаемое по модулю меньше точности. Требуется, во-первых, вывести на экран таблицу значений i и суммы i слагаемых. Во-вторых, построить график этой функции. По оси абсцисс выводить i , по оси ординат - сумму i слагаемых. Масштаб графика должен изменяться автоматически в зависимости от значения x .

Варианты

$$1. f(x) = \sum_{i=1}^{\infty} \frac{(-1)^i x^{i+1}}{(i+2)i!}.$$

$$2. f(x) = \sum_{i=1}^{\infty} \frac{(-1)^i x^i}{(i+2)!!}.$$

$$3. f(x) = \sum_{i=1}^{\infty} \frac{(-1)^i (-2x)^{i+1}}{(2i+2)^{2i}}.$$

$$4. f(x) = \sum_{i=1}^{\infty} \frac{x \cos x}{(i+1)^2}.$$

$$5. f(x) = \sum_{i=1}^{\infty} \frac{(-1)^{(i+1)} x^{2i+1}}{(3i+1)!i}.$$

$$6. f(x) = \sum_{i=1}^{\infty} \frac{(-2)^i x!}{(i+1)!i!}.$$

$$7. f(x) = \sum_{i=1}^{\infty} \frac{(-1)^i x^{2i-1}}{(\exp(i))i!}.$$

$$8. f(x) = \sum_{i=1}^{\infty} \frac{(-1)^{2i+1} x^{2i}}{(3i-5)!}.$$

$$9. f(x) = \sum_{i=1}^{\infty} \frac{\sin x * x^{i+1}}{(i-5)i^3}.$$

$$10. f(x) = \sum_{i=1}^{\infty} \frac{\cos(x+i)}{(i+4)(2i-3)}.$$

$$11. f(x) = \sum_{i=1}^{\infty} \frac{(-1)^i x^3}{(i+5)i^i}.$$

$$12. f(x) = \sum_{i=1}^{\infty} \frac{(-1)^i \sin x^2}{(i+1)!!}.$$

$$13. f(x) = \sum_{i=0}^{\infty} \frac{(-1)^i x^{4i+3}}{(2i+1)!(4i+3)}.$$

$$14. f(x) = \sum_{i=1}^{\infty} \frac{x \cos(2x+1)}{(2i+1)(2i-1)i!!}.$$

$$15. f(x) = \sum_{i=1}^{\infty} \frac{(2i)! + |x|}{(i^2)!}.$$

$$16. f(x) = \sum_{i=1}^{\infty} \frac{(-1)^{(i)} x^i}{(i!+1)!}.$$

$$17. f(x) = \sum_{i=1}^{\infty} \frac{(-1)^i x^{-2+i}}{(i+1)!!}.$$

$$18. f(x) = \sum_{i=1}^{\infty} \frac{(-1)^{i+1} 3x^4}{7i * i!!}.$$

$$19. f(x) = \sum_{i=0}^{\infty} \frac{(-1)^i x^{4i+1}}{(4i+1)(2i)!}.$$

$$20. f(x) = \sum_{i=0}^{\infty} \frac{(-1)^i (x/2)^{2i+1}}{i!(i+1)!}.$$

$$21. f(x) = \sum_{i=0}^{\infty} \frac{(-1)^{(i+1)} x^{2i-1}}{(2i+1)!(2i-1)}.$$

$$22. f(x) = \sum_{i=0}^{\infty} \frac{(-2)^{i+1} x^{2i}}{(i+1)!i^2}.$$

$$23. f(x) = \sum_{i=1}^{\infty} \frac{(-1)^{2i+1} x^i}{(i^3-3)!!}.$$

$$24. f(x) = \sum_{i=1}^{\infty} \frac{x}{i^3 + i\sqrt{|x|} + 1}.$$

$$25. f(x) = \sum_{i=1}^{\infty} \frac{(-1)^{(i+1)}}{(2i+1)!} \left(\frac{x}{3}\right)^{4i+2}.$$

$$28. f(x) = \sum_{i=0}^{\infty} \frac{x^{2i}}{i!2^i}.$$

$$26. f(x) = \sum_{i=0}^{\infty} \frac{(-1)^i x^i}{(5i+1)^3}.$$

$$29. f(x) = \sum_{i=1}^{\infty} \frac{x^i \cos(x+1)}{3i(2i-1)^4}.$$

$$27. f(x) = \sum_{i=0}^{\infty} \frac{(-1)^i (i+1)x^i}{(3i)^i}.$$

$$30. f(x) = \sum_{i=1}^{\infty} \frac{(-1)^{(i)} x^i}{(i^2 + 1)!}.$$

Лабораторная работа № 3

Цель работы. Решить задачу на обработку одномерного массива. Требуется ввести с клавиатуры размерность массива n , далее в цикле ввести элементы массива. Полученный ответ вывести на экран.

Варианты

1. Даны натуральные n, m , целые числа a_1, \dots, a_n и b_1, \dots, b_m . Внутри каждой из данных последовательностей нет повторяющихся членов. Построить объединение данных последовательностей.

2. Даны натуральные n, m , целые числа a_1, \dots, a_n и b_1, \dots, b_m . Внутри каждой из данных последовательностей нет повторяющихся членов. Построить пересечение данных последовательностей.

3. Даны натуральные n, m , целые числа a_1, \dots, a_n и b_1, \dots, b_m . Внутри каждой из данных последовательностей нет повторяющихся членов. Получить все члены последовательности a_1, \dots, a_n , которые не входят в последовательность b_1, \dots, b_m .

4. Даны натуральные n, m , целые числа a_1, \dots, a_n и b_1, \dots, b_m . Внутри каждой из данных последовательностей нет повторяющихся членов. Верно ли, что все члены последовательности a_1, \dots, a_n входят в последовательность b_1, \dots, b_m ?

5. Даны натуральные n, m , целые числа a_1, \dots, a_n и b_1, \dots, b_m . Внутри каждой из данных последовательностей нет повторяющихся членов. Верно ли, что все члены последовательности

a_1, \dots, a_n входят в последовательность b_1, \dots, b_m и при этом a_1 встречается в последовательности b_1, \dots, b_m не позднее, чем a_2 ; a_2 – не позднее, чем a_3 , и так далее?

6. Даны натуральное n , целые числа a_1, \dots, a_n . Внутри данной последовательности могут быть повторяющиеся члены. Получить все числа, которые входят в последовательность по одному разу.

7. Даны натуральное n , целые числа a_1, \dots, a_n . Внутри данной последовательности могут быть повторяющиеся члены. Получить числа, взятые по одному из каждой группы равных членов.

8. Даны натуральное n , целые числа a_1, \dots, a_n . Внутри данной последовательности могут быть повторяющиеся члены. Выяснить, сколько чисел входит в последовательность по одному разу.

9. Даны натуральное n , целые числа a_1, \dots, a_n . Внутри данной последовательности могут быть повторяющиеся члены. Найти число различных членов последовательности.

10. Даны натуральное n , целые числа a_1, \dots, a_n . Внутри данной последовательности могут быть повторяющиеся члены. Выяснить, сколько чисел входит в последовательность более чем по одному разу.

11. Даны натуральное n , целые числа a_1, \dots, a_n . Внутри данной последовательности могут быть повторяющиеся члены. Выяснить, имеется ли в последовательности хотя бы одна пара совпадающих чисел.

12. Даны натуральное n , целые числа a_1, \dots, a_n . Пусть M – наибольшее, а m – наименьшее из данной последовательности. Получить в порядке возрастания все целые из интервала (m, M) , которые не входят в последовательность a_1, \dots, a_n .

13. Даны натуральные n, m , целые числа a_1, \dots, a_n и b_1, \dots, b_m . Верно ли, что эти две последовательности отличаются не более чем порядком следования членов?

14. Даны натуральное n , целые числа a_1, \dots, a_n . Для каждого из чисел, входящего в последовательность, выяснить, сколько именно раз оно входит в эту последовательность.

15. Даны натуральное n , действительные числа x, a_1, \dots, a_n . В последовательности a_1, \dots, a_n найти два члена, среднее арифметическое которых ближе всего к x .

16. Даны натуральные n, m , целые числа a_1, \dots, a_n и b_1, \dots, b_m . Внутри каждой из данных последовательностей нет повторяю-

щихся членов. Верно ли, что все члены последовательности b_1, \dots, b_m входят в последовательность a_1, \dots, a_n ?

17. Даны натуральное n , целые числа a_1, \dots, a_n . Определить, сколько чисел последовательности отличных от a_n . Если это количество четное, то подсчитать сумму элементов массива, иначе произведение.

18. Даны натуральное n , целые числа a_1, \dots, a_n . Наименьший член последовательности заменить целой частью среднего арифметического всех членов, остальные члены оставить без изменения. Если в последовательности несколько членов со значением минимума, то заменить последний по порядку.

19. Даны натуральное n , целые числа a_1, \dots, a_n . Если в данной последовательности ни одно четное число не расположено после нечетного, то вывести на экран все отрицательные члены последовательности в обратном порядке, иначе – все положительные.

20. Даны натуральное n , целые числа a_1, \dots, a_n . Переставить члены последовательности так, чтобы сначала расположились все ее неотрицательные члены, а потом – отрицательные, причем в том же порядке, как они встречаются в исходной последовательности.

21. Даны натуральное n , целые числа a_1, \dots, a_n . Оставить без изменения данную последовательность, если она упорядочена по неубыванию или невозрастанию; в противном случае удалить из последовательности те члены, порядковые номера которых кратны четырем, сохранив прежним порядок оставленных членов.

22. Даны натуральные n, m , целые числа a_1, \dots, a_n . Найти три натуральных числа i, j, k , каждое из которых не превосходит n , такие что $a_i + a_j + a_k = m$. Если таких чисел нет, то сообщить об этом.

23. Даны натуральное n , целые числа a_1, \dots, a_n и b_1, \dots, b_n . Преобразовать последовательность b_i по правилу: если $a_i \leq 0$, то b_i увеличить в 10 раз, иначе b_i заменить нулем.

24. Даны натуральное n , целые числа a_1, \dots, a_n . Получить последовательность b_1, \dots, b_n , где b_i равно сумме тех членов данной последовательности, которые принадлежат отрезку $[i-5, i+5]$. Если отрезок не содержит членов последовательности, то соответствующее b_i положить равным нулю.

25. Даны натуральное n , целые числа a_1, \dots, a_n . Все числа последовательности попарно различны. Поменять в этой

последовательности местами наименьший и наибольший члены, последний член и первый отрицательный.

26. Даны натуральное n , целые числа a_1, \dots, a_n . Все числа последовательности попарно различны. Отсортировать последовательность по возрастанию и циклически сдвинуть ее вправо на два элемента.

Лабораторная работа № 4

Цель работы. Решить задачу на обработку текста, используя функции работы со строками. Всюду ниже, если это не оговорено особо, предполагается, что исходным является текстовый файл. В текст могут входить слова из латинских букв, цифры, знаки арифметических операций, точка, запятая, пробел. Требуется считать текст из файла, вывести его на экран, после решения задачи вывести на экран результат.

Варианты.

1. Вывести на печать все строки из этого файла, длина которых больше 15 символов, всюду заменяя 0 на *.

2. Дан текстовый файл, содержащий программу на языке паскаль. Проверить эту программу на соответствие открывающих и закрывающих скобок. Считать, что один оператор языка может занимать не более трех строк.

3. Вывести самую длинную строку файла в перевернутом виде. Если таких строк несколько – то первую.

4. Вывести на экран красным цветом ту строку, где максимальное количество попарно различных символов.

5. Вывести на экран красным цветом ту строку, где наименьшее число пробелов.

6. Вывести на экран красным цветом ту строку, где максимальное количество символов « h » и « m ».

7. Вывести на экран те строки, где встречается сочетание символов «,-» не менее двух раз.

8. Вывести на экран только буквы и знаки «.», «.», удалив из текста цифры и остальные символы.

9. В каждой строке заменить все маленькие буквы после пробела или точки на большие, и вывести полученный текст на экран.

10. Удалить из текста все символы, кроме букв и цифр, заменить все большие буквы на маленькие. Вывести полученный текст на экран.

11. Удалить из каждой группы цифр, которой не предшествует точка, все начальные нули (кроме последнего, если за ним – точка).

12. На экран вывести из каждой строки только те фрагменты текста, которые расположены между первым и вторым двоеточием, третьим и четвертым, пятым и шестым и т.д.

13. Найти число тех групп букв, которые заканчиваются той же буквой, что и первая группа букв.

14. Если в данном тексте есть группа знаков, начинающаяся знаком «*», то в следующей по порядку группе знаков (если она существует) заменить каждый из знаков нулем.

15. Если в тексте нет символа «+», то оставить текст без изменения, иначе каждую из цифр, предшествующую первому вхождению знака «+», заменить на «*».

16. Вывести на экран текст, составленный из последних букв всех слов данного файла.

17. Вывести на экран текст, составленный из всех слов данного файла, содержащих ровно две буквы «d».

18. Вывести на экран в алфавитном порядке все различные строчные буквы, входящие в текст из данного файла.

19. Вывести на экран те слова, которые отличны от последнего слова текста и первая буква этого слова входит в него еще раз.

20. Вывести на экран те слова, которые отличны от последнего слова первой строки текста и симметричны.

21. Вывести на экран все слова, которые отличны от последнего слова текста, предварительно преобразовав каждое из них по следующему правилу: из каждого слова нечетной длины удалить его среднюю букву.

22. Вывести на экран все слова, которые отличны от последнего слова текста, предварительно преобразовав каждое из них по следующему правилу: из каждого слова удалить все последующие вхождения его первой буквы.

23. Вывести на экран все слова, которые отличны от последнего слова текста, предварительно преобразовав каждое из

них по следующему правилу: из каждого слова удалить все предыдущие вхождения его последней буквы.

24. Вывести на экран все слова, которые отличны от последнего слова текста, предварительно преобразовав каждое из них по следующему правилу: оставить в слове только первые вхождения каждой буквы.

25. Вывести на экран все слова текста в алфавитном порядке.

26. Вывести на экран те слова, которые отличны от последнего слова текста и все буквы этого слова различны.

27. Вывести на экран все слова, содержащие наибольшее количество гласных латинских букв (a, e, i, u, o).

28. Для каждого из слов указать, сколько раз оно встречается в данном тексте.

29. Найти количество слов, у которых первый и последний символы совпадают. Вывести на экран все слова минимальной длины.

30. Даны натуральное число n , символ s ($n \leq 1000$, s – одна из букв и, р, д, в, т, п, указывающая падеж – именительный, родительный, дательный, винительный, творительный, предложный). Записать количественное числительное, обозначающее n , в соответствующем падеже.

Лабораторная работа № 5

Цель работы. Использование процедур и функций в задаче на обработку двумерного массива.

Требуется вывести на экран меню, состоящее из следующих пунктов:

- 1) ввод матрицы с клавиатуры,
- 2) ввод матрицы из файла,
- 3) вычисление характеристики,
- 4) преобразование матрицы,
- 5) печать матрицы,
- 6) выход.

и обеспечить его функционирование.

Внутри программы характеристика оформляется в виде функции с параметрами - значениями, которая возвращает значение булевского типа; преобразование в виде процедуры с

параметрами – переменными. Необходимо отслеживать, был ли произведен ввод данных до выбора пунктов меню, которые обрабатывают матрицу.

Варианты характеристик

1. В матрице существует строка, элементы которой образуют симметричную последовательность.

2. В строках с отрицательным элементом на главной диагонали найти максимум. Если максимум отрицательный или таких строк нет, ответом является число ноль, иначе единица.

3. Количество нулей в заштрихованной области совпадает с количеством нулей в остальной части матрицы (рис. 1, а).

4. Индексы минимального по модулю элемента матрицы четны.

5. Наименьший элемент матрицы расположен в заштрихованной области (рис. 1, б).

6. Максимальный элемент матрицы расположен в заштрихованной области (рис. 1, в).

7. Индексы минимального элемента в заштрихованной области четны (рис. 1, г).

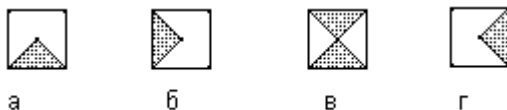


Рис. 1

8. Номер строки, все элементы которой образуют убывающую последовательность, нечетен.

9. В матрице существует столбец, все элементы которого образуют возрастающую последовательность.

10. В матрице существует строка, в которой элемент, расположенный на главной диагонали, равен максимальному из предшествующих элементов в этой строке.

11. В матрице число строк, в которых расположены минимальные элементы, четно.

12. Наибольший элемент матрицы расположен ниже побочной диагонали.

13. Индексы максимального элемента матрицы четны.

14. Каждая из строк матрицы с четным индексом содержит не более трех положительных элементов.

15. Количество отрицательных элементов выше главной диагонали равно количеству положительных элементов ниже ее.

16. Наибольший из минимальных элементов столбцов отрицателен.

17. Максимальные элементы каждой строки матрицы образуют убывающую последовательность.

18. Все строки матрицы с нечетными номерами содержат одинаковое количество нулей.

19. Наименьшее значение из элементов побочной диагонали и двух соседних с ней линий равно нулю.

20. Наибольшее значение элементов главной диагонали равно наибольшему значению элементов побочной диагонали.

21. Минимальный элемент матрицы расположен в заштрихованной области (рис. 1, в).

22. Средние арифметические значения элементов всех столбцов матрицы с нечетными номерами равны.

23. Индексы максимального элемента в заштрихованной области четны (рис. 1, г).

24. Минимальные элементы каждого столбца матрицы образуют симметричную последовательность.

25. В матрице количество строк с неотрицательными элементами четно.

26. В матрице количество столбцов, в которых есть два подряд идущих отрицательных элемента, нечетно.

27. В матрице существует строка, все элементы которой нечетные числа.

Варианты преобразований матрицы.

1. Повернуть внешний контур матрицы на 90^0 по часовой стрелке, второй контур на 90^0 против часовой стрелки.

2. Заменить нулями в матрице те элементы с четной суммой индексов, для которых имеются равные среди элементов первой строки.

3. Поменять местами в матрице строки, где расположены минимальный и максимальный элемент соответственно.

4. Удалить из матрицы строку и столбец, на пересечении которых расположен максимальный по модулю элемент.

5. Поменять местами строки в матрице: первую с n -ой, вторую – с $(n-1)$ -ой, и т.д.
6. Преобразовать матрицу в строку по спирали.
7. Получить матрицу, равную квадрату исходной.
8. Преобразовать матрицу в строку по схеме (рис. 2, а).
9. Выкинуть из матрицы столбец, обладающий наибольшей по модулю суммой элементов.
10. Преобразовать матрицу в строку по схеме (рис. 2, б).
11. Получить новую матрицу, переставляя блоки старой (рис. 2, в).

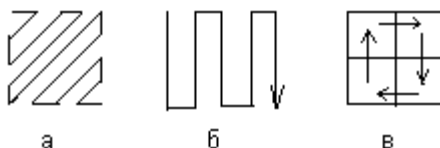


Рис. 2

12. Получить новую матрицу B , элемент b_{ij} которой равен наибольшему из элементов a_{kl} исходной матрицы, где k меняется от 1 до i , а l – от j до n .
13. Повернуть матрицу на 90° против часовой стрелки.
14. Повернуть матрицу на 180° по часовой стрелке.
15. Поменять в матрице местами заштрихованные блоки (рис. 1, в).
16. Поменять местами столбцы в матрице: первый с n -м, второй – с $(n-1)$ -м, и т.д.
17. Получить куб исходной матрицы.
18. Получить новую матрицу B , элемент b_{ij} которой равен наименьшему из элементов a_{kl} исходной матрицы, где k меняется от i до n , а l – от 1 до j .
19. Преобразовать матрицу в строку по схеме (рис. 3, а).
20. Повернуть второй контур матрицы на 180° .
21. Получить новую матрицу B , элемент b_{ij} которой равен произведению элементов a_{kl} исходной матрицы, где k меняется от i до n , а l – от 1 до j .
22. Преобразовать матрицу, переставляя местами ее блоки (рис. 3, б).

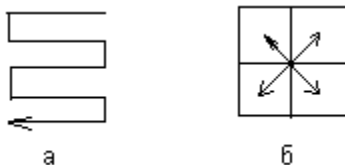


Рис. 3

23. Получить новую матрицу B , элемент b_{ij} которой равен произведению элементов a_{kl} исходной матрицы, где k и l меняются от $\min(i, j)$ до $\max(j, i)$.

24. Получить новую матрицу B , элемент b_{ij} которой равен сумме элементов a_{kl} исходной матрицы, где k меняется от 1 до i , а l – от j до n .

25. Получить новую матрицу B , элемент b_{ij} которой равен произведению элемента a_{ij} исходной матрицы на максимум i строки.

26. Получить новую матрицу B , элемент b_{ij} которой равен наибольшему из элементов a_{kl} исходной матрицы, где k меняется от 1 до i , а l – от j до n .

27. Получить новую матрицу B , элемент b_{ij} которой равен произведению элемента a_{ij} исходной матрицы на минимум j столбца.

Лабораторная работа № 6

В данном разделе собраны алгоритмически интересные задачи, решение которых требует творческого подхода и самостоятельной разработки как структур данных, так и алгоритма.

Варианты

1. Дан массив целых чисел $x[1]..x[m+n]$, рассматриваемый как соединение двух его отрезков: начала $x[1]..x[m]$ длины m и конца $x[m+1]..x[m+n]$ длины n . Не используя дополнительных массивов, переставить начало и конец. (Число действий порядка $m+n$.)

2. Некоторое число содержится в каждом из трех целочисленных неубывающих массивов $x[1] \leq \dots \leq x[p]$, $y[1] \leq \dots \leq y[q]$,

$z[1] \leq \dots \leq z[r]$. Найти одно из таких чисел. Число действий должно быть порядка $p + q + r$.

3. На шашечной доске размера $n \times n$ её верхний правый угол имеет номер $(1,1)$. В позиции (i,j) стоит фишка, которая может двигаться по правилу, показанному на рисунке 4 (допустимые ходы обозначены х, фишка - О). Фишка не может дважды ставиться на одно и то же поле.

Играют двое, по очереди двигая фишку. Выигрывает поставивший фишку в позицию $(1,1)$. Для введенных i, j определить, может ли выиграть первый игрок при наилучших ходах второго. Написать программу, где первый игрок - компьютер.

									1
									1
		х	х	х					
			О	х					
				х					
									n

Рис. 4

4. Есть две обезьяны и куча из L бананов. Обезьяны по очереди, начиная с первой, берут из кучи бананы, причем 1-ая обезьяна может при каждом очередном ходе взять из кучи либо a_1 , либо a_2 , либо ... a_s бананов ($a_1 < a_2 < \dots < a_s$), а 2-ая при каждом очередном ходе – либо b_1 , либо b_2 , либо ... b_k бананов ($b_1 < b_2 < \dots < b_k$). Нумерация индексов при a и b не имеет никакого отношения к номерам ходов обезьян. Выигрывает та обезьяна, которая на своем ходе не может взять банан(ы) (либо потому, что их не осталось, либо потому, что бананов осталось меньше чем a_1 (при ходе первой обезьяны), либо b_1 (при ходе второй обезьяны)).

Определить, может ли выиграть первая обезьяна при наилучших ходах соперницы, которая также стремится выиграть. Все входные данные - натуральные числа.

5. Пусть слово - это последовательность от 1 до 8 символов, не включающая пробелов.

Вводится n слов A_1, \dots, A_n . Можно ли их переупорядочить так, чтобы получилась "цепочка", т.е. для каждого слова A_j его первая буква должна совпадать с последней буквой предыдущего слова, а последняя буква в A_j - с первой буквой последующего слова; соответственно последняя буква последнего слова должна совпадать с первой буквой первого слова. В цепочку входят все n слов без повторов.

Дать ответ в виде "Можно"/"Нельзя". Если такое упорядочение возможно, то вывести какую-нибудь цепочку слов. Слова при выводе разделяются пробелами.

6. На линейке из m клеток в разных концах стоят две фишки, которые ходят по очереди. Каждая из фишек может ходить влево или вправо не более чем на k клеток ($m \leq 80$; $k \leq m-2$). При этом нельзя перешагивать через фишку и нельзя оставаться на месте. Фишка проигрывает, если она не может сделать ход. Написать программу, реализующую выигрышную стратегию для одной из фишек. При этом разрешается передача хода в самом начале игры. Предусмотреть контроль входных данных.

7. Составить программу, которая по введенному N выдает последовательность длины N , состоящую из цифр 0 и 1 такую, что ни один фрагмент этой последовательности не повторяется подряд трижды.

8. Даны три слова X, Y, Z . Определить, существует ли слово V такое, что X, Y, Z являются повторениями слова V . Если V существует, то напечатать его. Слова имеют длину не более 1000 символов. Символ "пробел" является разделителем слов.

9. Ввести три целых числа a, b, c .

1) Представить a в виде суммы минимального числа целых слагаемых $x[i]$, каждое из которых лежит на отрезке $[b, c]$.

2) Можно ли представить число a таким образом, чтобы

$$a = x[1] * x[2] * \dots * x[k] = \prod_{i=1}^k x[i], \text{ где } b \leq x[i] \leq c, x[i] - \text{целые.}$$

Лучшим считается алгоритм, находящий такое представление с наименьшим числом множителей. Предусмотреть вариант, когда такого представления не существует.

10. Пусть A - последовательность букв. После вычеркивания одной буквы из A (в одной позиции) получили последовательность B . После вычеркивания другой буквы из A (в одной позиции) получили последовательность C . Можно ли по последовательностям B и C :

- 1) Определить вычеркнутые буквы.
- 2) Определить последовательность A .

Примечание: B и C могут быть получены вычеркиванием одной и той же буквы.

11. Чтобы зашифровать текст из 121 буквы, его можно записать в квадратную матрицу порядка 11 по строкам, а затем прочитать по спирали, начиная с центра (т.е. с элемента с индексами 6,6). Написать программу шифровки и дешифровки. Текст вводить из файла.

12. Шифровка текста с помощью решетки заключается в следующем. Решетка, т.е. квадрат 10×10 клеток, некоторые клетки в котором вырезаны, совмещается с целым квадратом 10×10 клеток и буквы, видимые через прорезы, переносятся в результирующую строку. Затем решетка поворачивается на 90° и считываются следующие буквы. И так еще два раза. Решетка изображается матрицей из нулей и единиц (ноль – прорезь), причем только один из элементов $a[i, j]$, $a[10-i+1, j]$, $a[i, 10-j+1]$, $a[10-i+1, 10-j+1]$ равен нулю. Написать программу шифровки и дешифровки. Текст вводить из файла.

13. На окружности задано $2n$ точек, пронумерованных от 1 до $2n$. Перечислить все способы провести n непересекающихся хорд с вершинами в этих точках.

14. Перечислить все способы разрезать n -угольник на треугольники, проведя $n - 2$ его диагонали.

15. Дана действительная матрица размером $m \times n$. Пусть для каждой строки матрицы определена функция $f(a)$, равная сумме нечетных элементов строки a . Упорядочить (переставить) строки матрицы по неубыванию в соответствии со следующим критерием: строка a считается "больше" строки b , если $f(a) > f(b)$. Используйте алгоритм простого выбора.

16. Счастливые билеты. Последовательность из $2n$ цифр (каждая цифра от 0 до 9) называется счастливым билетом, если

сумма первых n цифр равна сумме последних n цифр. Найти число счастливых последовательностей данной длины.

17. Лабиринт задается матрицей смежности C размера $N \times N$, где $C(i, j)=1$, если узел i связан узлом j посредством дороги. Часть узлов назначается входами, часть - выходами. Входы и выходы задаются последовательностями узлов $X(1), \dots, X(p)$ и $Y(1), \dots, Y(k)$ соответственно. Найти максимальное число людей, которых можно провести от входов до выходов таким образом, чтобы их пути не пересекались по дорогам, но могут пересекаться по узлам;

18. Лабиринт задается матрицей смежности C размера $N \times N$, где $C(i, j)=1$, если узел i связан узлом j посредством дороги. Часть узлов назначается входами, часть - выходами. Входы и выходы задаются последовательностями узлов $X(1), \dots, X(p)$ и $Y(1), \dots, Y(k)$ соответственно. Найти максимальное число людей, которых можно провести от входов до выходов таким образом, чтобы их пути не пересекались по узлам;

19. N шестеренок пронумерованы от 1 до N ($N \leq 10$). Заданы m ($0 \leq m \leq 45$) соединений пар шестеренок в виде (i, j) , $1 \leq i < j \leq N$ (шестерня с номером i находится в зацеплении с шестерней j). Можно ли повернуть шестерню с номером 1?

Если да, то найти количество шестерен, пришедших в движение.

Если нет, то требуется убрать минимальное число шестерен так, чтобы в оставшейся системе при вращении шестерни 1 во вращение пришло бы максимальное число шестерен. Указать номера убранных шестерен (если такой набор не один, то любой из них) и количество шестерен, пришедших в движение.

20. Имеется N прямоугольных конвертов и N прямоугольных открыток различных размеров. Можно ли разложить все открытки по конвертам, чтобы в каждом конверте было по одной открытке.

Замечание. Открытки нельзя складывать, сгибать и т.п., но можно помещать в конверт под углом. Например, открытка с размерами сторон 5:1 помещается в конверты с размерами 5:1, 6:3, 4.3:4.3, но не входит в конверты с размерами 4:1, 10:0.5, 4.2:4.2.

21. Составить программу для нахождения произвольного разбиения 20 студентов на 2 команды, численность которых отличается не более чем в 2 раза, если известно, что в любой

команде должны быть студенты, обязательно знакомые друг с другом. Круг знакомств задается матрицей A размера 20×20 с элементами $a_{ij} = \{1, \text{если } i \text{ студент знаком с } j; 0, \text{иначе}\}$.

22. Имеется N человек и прямоугольная таблица $A[1:N, 1:N]$; элемент $A[i, j]$ равен 1, если человек i знаком с человеком j , $A[i, j] = A[j, i]$. Можно ли разбить людей на 2 группы, чтобы в каждой группе были только незнакомые люди.

23. Имеется N городов. Для каждой пары городов (I, J) можно построить дорогу, соединяющую эти два города и не заходящие в другие города. Стоимость такой дороги $A(I, J)$. Вне городов дороги не пересекаются.

Написать алгоритм для нахождения самой дешевой системы дорог, позволяющей попасть из любого города в любой другой. Результаты задавать таблицей $B[1:N, 1:N]$, где $B[I, J] = 1$ тогда и только тогда, когда дорогу, соединяющую города I и J , следует строить.

24. Написать программу, которая вычисляет ранг данной целочисленной матрицы.

25. Написать программу нахождения обратной матрицы к данной. Элементами исходной матрицы являются действительные числа.

26. Даны три пробирки по 100 мл. Одна имеет n делений a_1, \dots, a_n , вторая и третья без делений. Третья пробирка полностью заполнена жидкостью. Составить программу, которая определяет, можно ли получить в одной из пробирок 1 мл, если да, привести последовательность переливаний.

27. Игра «Быки и коровы». Программа выбирает с помощью датчика случайных чисел четырехзначное число с разными цифрами. Нужно угадать это число. На каждом шаге играющий называет четырехзначное число, а программа сообщает, сколько угаданных цифр стоит на нужном месте («быки») и сколько угаданных цифр стоит не на своем месте («коровы»). Например, если программой загадано число 3572, а играющий назвал 2503, то он получит ответ: «1 бык, 2 коровы».

28. Игра «Виселица». Программа выбирает слово и рисует на экране столько прочерков, сколько букв в этом слове. Играющий должен отгадать это слово. В каждый ход играющий указывает одну букву. Если названа буква, входящая в состав слова, она

подставляется вместо соответствующего прочерка. В противном случае играющий теряет одно очко. В начальный момент времени у играющего 15 очков.

29. Написать программу игры «Морской бой».

30. Поле шахматной доски задается парой натуральных чисел: первое указывает номер вертикали при счете слева направо, второе – номер горизонтали при счете снизу вверх. Расстановка фигур задается таким образом, что вначале указывается поле, на котором стоит белый король, затем поля, на которых черные король, конь, слон. Охарактеризовать положение белых с помощью слов: шах, мат, пат, обыкновенная позиция. Вывести перечень позиций, куда может сделать ход белый король.

31. Вводится N - количество домов и k - количество дорог. Дома пронумерованы от 1 до N . Каждая дорога определяется тройкой чисел - двумя номерами домов - концов дороги и длиной дороги. В каждом доме живет по одному человеку. Найти точку - место встречи всех людей, от которой суммарное расстояние до всех домов будет минимальным.

Если точка лежит на дороге, то указать номера домов - концов этой дороги и расстояние от первого из этих домов. Если точка совпадает с домом, то указать номер этого дома. Примечание: длины дорог - положительные целые числа.

32. N колец сцеплены между собой (задана матрица $A(n \times n)$, $A(i, j)=1$ в случае, если кольца i и j сцеплены друг с другом и $A(i, j)=0$ иначе). Удалить минимальное количество колец так, чтобы получилась цепочка.

33. Дан файл, содержащий конечное множество русских слов. Вывести на экран все возможные кроссворды, составленные из этих слов.

34. Задано на плоскости множество из N прямоугольников, стороны которых параллельны осям координат, при этом каждый прямоугольник задается координатами левой нижней и правой верхней его вершин. Составить алгоритм определения наибольшего натурального числа K , для которого существует точка плоскости, принадлежащая одновременно K прямоугольникам.

Примечание: эффективным считается алгоритм, число действий которого пропорционально N^2 .

35. На квадратном торте N свечей. Можно ли одним прямым линейным разрезом разделить его на две равные по площади части, одна из которых не содержала бы ни одной свечи? Свечи будем считать точками, у которых известны их целочисленные координаты $x[1], y[1]; \dots; x[N], y[N]$ (начало координат - в центре торта); разрез не может проходить через свечу.

36. На решетке размера $m \times n$ заданы k точек своими координатами. Необходимо определить, можно ли построить выпуклый многоугольник такой, что каждая точка принадлежит некоторой стороне.

37. N точек на плоскости заданы своими координатами. Найти порядок, в котором можно соединить эти точки, чтобы получился N -угольник (т.е. не было бы пересечений сторон).

38. Многоугольник на плоскости задается координатами своих N вершин в порядке обхода их по контуру по часовой стрелке. Считается, что контур самопересечений не имеет. Найти площадь, периметр и углы многоугольника.

39. Определить количество повторений каждой из цифр 0, 1, 2, ..., 9 в числе N^N (N в степени N), $N \leq 1000$.

40. Натуральное число $N > 1$ представить в виде суммы натуральных чисел так, чтобы произведение этих слагаемых было максимально.

II. Языки программирования

Лабораторная работа № 1. Внутреннее представление целочисленных данных в IBM PC

Цель работы. Выполнить перевод заданных преподавателем чисел из десятичной в двоичную систему счисления. Дать их внутреннее (машинное) представление в соответствии с диапазоном в знаковых и беззнаковых форматах типов **ShortInt (signed char)**, **Byte (unsigned char)**, **Integer (int)**, **Word (unsigned int)**, **LongInt (int)**. Машинное представление данных должно быть дано в двоичной и шестнадцатичной системах счисления.

Порядок работы.

1. Вычислить для своего варианта целые числа.
2. Перевести их из десятичной в двоичную (или шестнадцатичную) систему счисления.
3. Получить их внутреннее представление.
4. Написать программу описания этих чисел на языке Ассемблера и получить листинг.
5. Проверить правильность своих выкладок.

В отчете по лабораторной работе должен быть представлен подробный протокол перевода всех заданных чисел из десятичной в двоичную и шестнадцатичную систему счисления.

Варианты

Преподаватель задает два базовых числа $\pm X$ и $\pm Y$. Студент должен прибавить и отнять от них № своего варианта.

Например, пусть $X = \pm 6534$, $Y = \pm 70$, № = 23. Тогда получаются следующие восемь целых чисел для варианта № = 23, а именно:

- | | |
|---------------------------|-----------------------|
| 1. $6534 + 23 = 6557$; | 5. $70 + 23 = 93$; |
| 2. $6534 - 23 = 6511$; | 6. $70 - 23 = 47$; |
| 3. $-6534 + 23 = -6511$; | 7. $-70 + 23 = -47$; |
| 4. $-6534 - 23 = -6557$; | 8. $-70 - 23 = -93$. |

Пример решения.

Протокол перевода чисел.

47	2				
46	23	2			
1	22	11	2		
	1	10	5	2	
		1	4	2	2
			1	2	1
				0	

47d → 00101111b

- 47d → 1) 00101111b – двоичный код числа |-47|

2) 11010000b – инверсия

3) + 1

11010001b – дополнительный код

Остальные числа аналогично.

Машинное представление.

Dec	Byte		Word	
	Bin	Hex	Bin	Hex
47	00101111	2F	0000000000101111	002F
- 47	11010001	D1	1111111111010001	FFD1
93	01011101	5D	0000000001011101	005D
-93	10100011	A3	1111111110100011	FFA3
6511			0001000110011111	119F
-6511			1110111001100001	EE61
6557			0001000111001101	11CD
-6557			1110111000110011	EE33

Файл листинга

Turbo Assembler Version 3.2 23/03/05 09:35:48 Page 1
pervod.asm

```
1 0000      data      segment para public 'data'
2              ; Byte
3 0000  2F              db  47
4 0001  5D              db  93
5              ; Word
6 0002  002F            dw  47
7 0004  005D            dw  93
8 0006  119F            dw 4511
9 0008  11CD            dw 4557
10             ; ShortInt
11 000A  2F              db  47
12 000B  5D              db  93
13 000C  D1              db -47
14 000D  A3              db -93
15             ; Integer
16 000E  002F            dw  47
17 0010  005D            dw  93
18 0012  119F            dw 4511
19 0014  11CD            dw 4557
20 0016  FFD1            dw -47
21 0018  FFA3            dw -93
22 001A  EE61            dw -4511
23 001C  EE33            dw -4557
24             ; Longint
25 001E  0000002F        dd  47
26 0022  0000005D        dd  93
27 0026  0000119F        dd 4511
28 002A  000011CD        dd 4557
29 002E  FFFFFFFD1       dd -47
30 0032  FFFFFFFA3       dd -93
31 0036  FFFFFFFE61      dd -4511
32 003A  FFFFFFFE33      dd -4557
33 003E              data      ends
34                  end
```

Лабораторная работа № 2. **Вычисление целочисленных** **арифметических выражений**

Цель работы. Вычислить заданное целочисленное выражение для исходных данных в знаковых форматах длиной 8 и 16 бит: **ShortInt (signed char)** и **Integer (int)**, используя арифметические операции ADD, ADC, INC, SUB, SBB, DEC, NEG, IMUL, IDIV, CBW, CWD и, если нужно, логические операции XOR, SAL, SAR. Исходные данные задаются в сегменте данных. Результат хранится в сегменте данных.

Варианты

- | | |
|-------------------------------------|------------------------------------|
| 1. $(2*c - d + 23)/(a/4 - 1);$ | 16. $(c + 4*d - 123)/(1 - a/2);$ |
| 2. $(-2*c + d*82)/(a/4 - 1);$ | 17. $(2*c + d - 52)/(a/4 + 1);$ |
| 3. $(c/4 - d*62)/(a*a + 1);$ | 18. $(-2*c - d + 53)/(a/4 - 1);$ |
| 4. $(b/2 - 53/c)/(b - a*c + 1);$ | 19. $(c*4 + 28*d)/(a*d - c - 1);$ |
| 5. $(25 + 2*a/c)/(c*a - b - 1);$ | 20. $(c + 23 - b*4)/(a*c/b - 1);$ |
| 6. $(c - d/2 + 33)/(2*a*a - 1);$ | 21. $(4*c - d/2 + 23)/((a*a - 1);$ |
| 7. $(c*d + 23)/(a/2 - 4*d - 1);$ | 22. $(c/d + 3*a/2)/(c - a - 1);$ |
| 8. $(-25/a + c - b*a)/(1 + c*b/2);$ | 23. $(2*c/a - d*d)/(d + a - 1);$ |
| 9. $(8*b + 1 - c)/(a/2 + b*c);$ | 24. $(4*a - b - 1)/(c/b + a);$ |
| 10. $(21 - a*c/4)/(1 + c/a + b);$ | 25. $(c - 33 + b/4)/(a*c/b - 1);$ |
| 11. $(a*b/4 - 1)/(41 - b*a + c);$ | 26. $(1 + 6*a - b/2)/(c + a/d);$ |
| 12. $(4*b/c + 1)/(2*a + a*c - b);$ | 27. $(b - b/c + a/4)/(a*b - 1);$ |
| 13. $(c/4 - d*42)/(c + a - 1);$ | 28. $(25/c - d + 2)/(b + a*a -$ |
| 14. $(a + b/4 - 1)/(c/3 - a*b);$ | 29. $(b*a + c/2)/(4*c - b + 1);$ |
| 15. $(a - c*b + 28)/(4*b/a + 10);$ | 30. $(c*b - 24 + a)/(b/2*c - 1).$ |

Лабораторная работа № 3. **Организация условных переходов**

Цель работы. Вычислить заданное условное целочисленное выражение для данных в форматах **ShortInt (signed char)** и **Byte (unsigned char)**, используя команды сравнения, условного и безусловного перехода. Исходные данные вводятся корректно (с проверкой на область допустимых значений). Результат **X** выводится на экран.

Варианты

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

$$2. \quad X = \begin{cases} (a-b)/a-31, & \text{если } a > b, \\ 2, & \text{если } a = b, \\ (a^3+1)/b, & \text{если } a < b; \end{cases}$$

$$3. \quad X = \begin{cases} a/b+21, & \text{если } a > b, \\ -82, & \text{если } a = b, \\ (a-b)/a, & \text{если } a < b; \end{cases}$$

$$4. \quad X = \begin{cases} b/a-1, & \text{если } a > b, \\ -120, & \text{если } a = b, \\ (a*b-9)/b, & \text{если } a < b; \end{cases}$$

$$5. \quad X = \begin{cases} (a/b-1)*a, & \text{если } a > b, \\ 125, & \text{если } a = b, \\ (a^3-b)/a, & \text{если } a < b; \end{cases}$$

$$6. \quad X = \begin{cases} b/a-2, & \text{если } a > b, \\ -25, & \text{если } a = b, \\ (a-100)/b, & \text{если } a < b; \end{cases}$$

$$7. \quad X = \begin{cases} a/b-2, & \text{если } a > b, \\ -7, & \text{если } a = b, \\ (a^3-5)/a, & \text{если } a < b; \end{cases}$$

$$8. \quad X = \begin{cases} (a-b)/a, & \text{если } a > b, \\ -2, & \text{если } a = b, \\ (b-45)/a, & \text{если } a < b; \end{cases}$$

$$9. \quad X = \begin{cases} 4/a+5, & \text{если } a > b, \\ 28, & \text{если } a = b, \\ (a^2-9)/b-2, & \text{если } a < b; \end{cases}$$

$$10. \quad X = \begin{cases} a/4-b, & \text{если } a > b, \\ -57, & \text{если } a = b, \\ (a^3-5)/b, & \text{если } a < b; \end{cases}$$

$$\begin{aligned}
11. \quad X &= \begin{cases} a/b - 42, & \text{если } a > b, \\ -11, & \text{если } a = b, \\ (a^2 - 8)/a, & \text{если } a < b; \end{cases} \\
12. \quad X &= \begin{cases} b/a - 121, & \text{если } a > b, \\ -117, & \text{если } a = b, \\ (3*a - 9)/b, & \text{если } a < b; \end{cases} \\
13. \quad X &= \begin{cases} a/b - 7 + 1, & \text{если } a > b, \\ -95, & \text{если } a = b, \\ (b^3 + 5)/a, & \text{если } a < b; \end{cases} \\
14. \quad X &= \begin{cases} (a - 4)/(a + 1), & \text{если } a > b, \\ -55, & \text{если } a = b, \\ (b^2 - 5)/b, & \text{если } a < b; \end{cases} \\
15. \quad X &= \begin{cases} b/a + 5, & \text{если } a > b, \\ -5, & \text{если } a = b, \\ (a*a - b)/b, & \text{если } a < b; \end{cases} \\
16. \quad X &= \begin{cases} a/b + 10, & \text{если } a > b, \\ -51, & \text{если } a = b, \\ (a*b - 4)/a, & \text{если } a < b; \end{cases} \\
17. \quad X &= \begin{cases} 52*b/a + b, & \text{если } a > b, \\ -125, & \text{если } a = b, \\ (a - 5)/b, & \text{если } a < b; \end{cases} \\
18. \quad X &= \begin{cases} (a*b - 1)/a, & \text{если } a > b, \\ 105, & \text{если } a = b, \\ (a^2 - 5)/b, & \text{если } a < b; \end{cases} \\
19. \quad X &= \begin{cases} 1 - b/a, & \text{если } a > b, \\ -10, & \text{если } a = b, \\ (b^2 - 5)/3, & \text{если } a < b; \end{cases} \\
20. \quad X &= \begin{cases} (a/b + 31)/a, & \text{если } a > b, \\ -25, & \text{если } a = b, \\ (5*b - 1)/a, & \text{если } a < b; \end{cases}
\end{aligned}$$

$$21. \quad X = \begin{cases} (2-b)/a, & \text{если } a > b, \\ -2, & \text{если } a = b, \\ (a/b-5)/b, & \text{если } a < b; \end{cases}$$

$$22. \quad X = \begin{cases} (a-b)/a, & \text{если } a > b, \\ 23, & \text{если } a = b, \\ (a^3-5)/b^2, & \text{если } a < b; \end{cases}$$

$$23. \quad X = \begin{cases} (3*a-b)/5, & \text{если } a > b, \\ -5, & \text{если } a = b, \\ (a^3-b)/b, & \text{если } a < b; \end{cases}$$

$$24. \quad X = \begin{cases} b/(a^2-1), & \text{если } a > b, \\ -78, & \text{если } a = b, \\ (a-98)/b, & \text{если } a < b; \end{cases}$$

$$25. \quad X = \begin{cases} 2*a/b+1, & \text{если } a > b, \\ -105, & \text{если } a = b, \\ (b+5)/a^3, & \text{если } a < b; \end{cases}$$

$$26. \quad X = \begin{cases} (a-b/a)*10, & \text{если } a > b, \\ a+25, & \text{если } a = b, \\ (a*b-5)/b, & \text{если } a < b; \end{cases}$$

$$27. \quad X = \begin{cases} (a-b)/(a+1), & \text{если } a > b, \\ a/25, & \text{если } a = b, \\ (2*a-5)/b, & \text{если } a < b; \end{cases}$$

$$28. \quad X = \begin{cases} (a*a-b)/2, & \text{если } a > b, \\ -a, & \text{если } a = b, \\ (a*b-5)/b, & \text{если } a < b; \end{cases}$$

$$29. \quad X = \begin{cases} (b+1)/a+1, & \text{если } a > b, \\ -b, & \text{если } a = b, \\ (a-5)/b^2, & \text{если } a < b; \end{cases}$$

$$30. \quad X = \begin{cases} (a-b^2)/a, & \text{если } a > b, \\ 25-a, & \text{если } a = b, \\ (3*a-5)/b, & \text{если } a < b; \end{cases}$$

Лабораторная работа № 4

Трансляция различных языковых конструкций в промежуточное представление

Цель работы. Написать программу, которая читает из текстового файла некоторую языковую конструкцию (оператор присваивания, условный оператор, оператор цикла или простую программу) и выполняет необходимые преобразования исходных данных в промежуточную форму, указанную в задании. При вводе данных из текстового файла необходимо выполнить синтаксический контроль.

Варианты

1. Заданное простое выражение, не содержащее вызовов функций и обращений к элементам массивов, преобразовать в обратную польскую запись. По полученной обратной польской записи построить трехадресную модель.

2. Построить график функции $y = F(x)$ на отрезке $[a, b]$. $F(x)$ вводится с клавиатуры и может содержать вызовы функций. Например, $y = x^2 * \sin(x)$, $y = 2x^2 + 5x - 4$ и т. д. Набор возможных функций, используемых в выражении, ограничен.

3. Заданное простое выражение, содержащее вызовы функций, преобразовать в обратную польскую запись. Например, $y = f(I, J + g(x + y), K * L, x(I, J), 5) + 2x$.

4. Построить обратную польскую запись для условного оператора (возможно вложение одного условного оператора в другой. Например, `if A then B else if C then K;`)

5. Заданное простое выражение, не содержащее вызовов функций и обращений к элементам массивов, преобразовать в обратную польскую запись. По полученной обратной польской записи построить одноадресную модель.

6. Заданное простое выражение, не содержащее вызовов функций и обращений к элементам массивов, преобразовать в прямую польскую запись.

7. Восстановить простое выражение по заданной его обратной польской записи.

8. Восстановить простое выражение по заданной его прямой польской записи.

9. Задана обратная польская запись некоторого простого выражения, не содержащего вхождений идентификаторов. Вычислить значение этого выражения.

10. Задана прямая польская запись некоторого простого выражения, не содержащего вхождений идентификаторов. Вычислить значение этого выражения.

11. Определить, эквивалентны ли две заданные логические формулы.

12. По заданной логической формуле построить эквивалентную логическую формулу, в которой знак отрицания встречается только перед переменными. Указания.

Воспользоваться эквивалентными преобразованиями:

$\text{NOT NOT переменная} \rightarrow \text{переменная}$

$\text{NOT(формула1 OR формула2)} \rightarrow \text{NOT (формула1) AND NOT(формула2)}$

$\text{NOT(формула1 AND формула2)} \rightarrow \text{NOT (формула1) OR NOT(формула2)}$

13. Пару скобок в логической формуле назовем *избыточной*, если после ее удаления формула останется эквивалентной исходной. Удалить все избыточные пары скобок в заданной формуле.

14. В заданной линейной программе удалить первый оператор каждой пары следующих друг за другом операторов в случае, если их левые части совпадают, а правая часть второго оператора не содержит вхождений переменной, совпадающей с его левой частью.

15. Вхождение оператора А линейной программы назовем избыточным, если среди предшествующих ему имеется еще одно вхождение этого оператора, причем левые части всех операторов, находящихся между двумя этими вхождениями, а также левая часть самого оператора не встречаются в правой части оператора А. Например, второе вхождение оператора $x:=y+z$ в линейной программе « $x:=y+z$; $u:=u-z$; $x:=y+z$;» избыточно. В заданной линейной программе удалить все избыточные.

16. Вхождение оператора А в линейную программу называется несущественным, если среди следующих после него операторов имеется еще одно вхождение оператора, скажем В, с то й же переменной - левой частью, что и у А, причем эта

переменная не встречается в правой части ни одного из операторов, находящихся между А и В. Например, Вхождение оператора $x:=u+v$ в линейную программу « $x:=u+v$; $u:=u*y$; $x:=y-u$ » несущественное. Удалить вхождения всех несущественных операторов из заданной линейной программы.

17. По заданной линейной программе, в которой последний оператор содержит в левой части переменную RESULT, построить простое выражение, значение которого совпадает с тем значением, которое получает переменная RESULT после исполнения линейной программы.

18. Упростить заданную правдоподобную простую программу, заменяя всякий оператор IF *константа1* *знак-отношения* *константа2* THEN *оператор* на *оператор* в случае истинности условия или удаляя этот условный оператор в случае ложности условия.

19. Проверить правдоподобность заданной простой программы. Простая программа содержит раздел описания переменных, оператор присваивания, составной оператор, короткий условный оператор. Простая программа называется правдоподобной, если выполнены следующие условия:

а) каждая из переменных, встречающихся в последовательности операторов тела программы, описана, т.е. встречается в списке переменных программы, причем ровно один раз;

б) в программе нет больше ни одной переменной, совпадающей с названием программы;

в) каждому использованию переменной текстуально предшествует хотя бы один оператор присваивания этой переменной или оператор ввода этой переменной.

20. Выполнить заданную линейную программу для заданных начальных значений всех встречающихся в ней переменных. Определить значения этих переменных после выполнения линейной программы.

21. Из-за недосмотра программиста был утерян раздел описания переменных правдоподобной простой программы (смотри задание 19). Восстановить его.

22. Реализовать упрощение логических формул относительно правил преобразования, заданных следующих схемами правил:

TRUE OR формула \rightarrow TRUE

TRUE AND формула \rightarrow формула

формула OR TRUE \rightarrow TRUE

формула AND TRUE \rightarrow формула

NOT FALSE \rightarrow TRUE

NOT TRUE \rightarrow FALSE

23. Реализовать упрощение логических формул относительно правил преобразования, заданных следующих схемами правил:

формула OR формула \rightarrow формула

формула AND формула \rightarrow формула

FALSE OR формула \rightarrow формула

формула OR FALSE \rightarrow формула

FALSE AND формула \rightarrow FALSE

формула AND FALSE \rightarrow FALSE

24. Составить таблицу истинности для заданной логической формулы.

25. Заданное простое выражение, содержащее вызовы функций и обращения к элементам массивов, преобразовать в обратную польскую запись. Например, $y=f(A[I, J+g(x+y), K*L], x(I, J), 5)$.

26. Что будет выведено на экран в результате выполнения правдоподобной простой программы?

27. Построить обратную польскую запись для оператора цикла.

Приложение 1.

Команды ассемблера

РЕЖИМЫ АДРЕСАЦИИ, ОПРЕДЕЛЯЕМЫЕ ПОСТБАЙТОМ

MOD R/M	00	01	10	11 (W=0)	11 (W=1)
000	[BX+SI]	[BX+SI]+D8	[BX+SI]+D1 6	AL	AX
001	[BX+DI]	[BX+DI]+D8	[BX+DI]+D1 6	CL	CX
010	[BP+SI]	[BP+SI]+D8	[BP+SI]+D1 6	DL	DX
011	[BP+DI]	[BP+DI]+D8	[BX+SI]+D1 6	BL	BX
100	[SI]	[SI]+D8	[SI]+D16	AH	SP
101	[DI]	[DI]+D8	[DI]+D16	CH	BP
110	D16	[BP]+D8	[BP]+D16	DH	SI
111	[BX]	[BX]+D8	[BX]+D16	BH	DI

00	ES	ПРИМЕЧАНИЕ
01	CS	W=0 байт, W=1 слово; D=1 в регистр, D=0 из регистра; D`=1-D; V=1 счетчик в регистре, V=0 счетчик непосредственный операнд
10	SS	R/M-регистр или память; REG-общий регистр; SEG-сегментный регистр S=0 расширение (8->16) не производится; S=1 расширение производится
11	DS	D8 непосредственный операнд байт, D16 непосредственный операнд слово

Приложение 2.

Структура программы

```
TITLE < Разместите здесь заголовок>
PAGE      ,132
STAK      SEGMENT PARA STACK 'STACK'
DB        64 DUP ('STACK')
STAK      ENDS

DATA      SEGMENT PARA PUBLIC 'DATA'
;<Поместите здесь данные >
DATA      ENDS

;<Если требуется оператор EXTRN, поместите его здесь>
;<Если требуется оператор PUBLIC, поместите его здесь>

CODE      SEGMENT PARA PUBLIC 'CODE'
START: ASSUME CS:CODE, SS:STAK, DS:DATA
MOV       AX, DATA
MOV       DS, AX

;<Поместите здесь команды>

MOV       AH, 4CH
INT       21H
CODE      ENDS
END       START
```

Создание объектного модуля

```
TASM /ZI ИМЯ.ASM, , ,
```

Создание загрузочного модуля

```
TLINK /V ИМЯ.OBJ+ ИМЯ1.OBJ+ ИМЯ2.OBJ
```

Запуск отладчика

```
TD ИМЯ.EXE
```

Пример макробιβλιοтеки

```
init macro
```

```
; Макрос настройки
```

```
    assume    cs:code, ss:stac, ds:data
```

```
    mov ax,data
```

```
    mov ds,ax
```

```
endm
```

```
vvod_str macro str
```

```
;ввод строки с клавиатуры
```

```
    push ax          ; строка описана в сегменте данных
```

```
    push dx          ; например str1 db 10, 11 dup (?)
```

```
    lea dx, str       ; 10 – максимальная длина
```

```
    mov ah, 0ah       ; по адресу str1+1 запишется текущая длина
```

```
    int 21h          ; с адреса str+2 - строка
```

```
    pop dx
```

```
    pop ax
```

```
endm
```

10			'П'	'р'	'и'	'в'	'е'	'т'				
----	--	--	-----	-----	-----	-----	-----	-----	--	--	--	--

```
out_str macro str
```

```
;Вывод строки str на экран
```

```
    push ax          ; строка описана в сегменте данных
```

```
    push dx          ; например
```

```
    lea dx, str       ; str1db 10, 13,'Введите матрицу',10,13,'$'
```

```
    mov ah, 09h
```

```
    int 21h
```

```
    pop dx
```

```
    pop ax
```

```
endm
```

```
clear macro
```

```
;очистка экрана
```

```
    push ax
```

```
    push bx
```

```
    push cx
```

```
    push dx
```

```
    mov ax, 0600h
```

```

mov bh, 07
mov cx, 0000
mov dx, 184fh
int 10h
pop dx
pop cx
pop bx
pop ax
endm

```

kursor macro stroka, stolbec

; Установка курсора в заданную позицию (stroka, stolbec)

```

push ax
push dx
push bx
mov ah, 2
mov bh, 0
mov dh, stroka
mov dl, stolbec
int 10h
pop bx
pop dx
pop ax
endm

```

vyv_matr macro matr, razmern, razmerm

; Вывод матрицы на экран

```

local cikl, cikl2
push ax
push bx
push cx
push si
push dx
push bp
mov bh, 0
mov ah, 03 ; чтение положения курсора
int 10h ;dh – строка, dl – столбец.
inc dh

```

```

mov     dl,8
cursor  dh, dl    ; перевод курсора на следующую строку
mov     cx, razmern
mov     bp,0      ;bp –смещение по строке
mov     si,0      ;si – смещение по столбцу
cikl1:  push     cx      ; цикл по строкам матрицы
        mov     cx, razmern
cikl2:  mov     al, matr[bp][si]; цикл вывода строки матрицы
        cbw
        call    write    ; вывод элемента строки
        inc     si      ; переход к следующему элементу строки
матрицы
        add     dl, 4
        cursor  dh, dl    ; перевод курсора для вывода следующего
        loop    cikl2    ; элемента строки
        add     bp, si    ; переход к следующей строке матрицы
        mov     si,0
        inc     dh
        mov     dl,8
        cursor  dh, dl    ; перевод курсора на следующую строку
        pop     cx
        loop    cikl1
        pop     bp
        pop     dx
        pop     si
        pop     cx
        pop     bx
        pop     ax
        endm

```

```

v_matr macro    matr, razmern, razmern

```

```

; Ввод матрицы

```

```

    local    cikl, cikl2
    push     ax
    push     bx
    push     cx
    push     si
    push     dx

```

```

push    bp
mov     bh, 0
mov     ah, 03      ; чтение положения курсора
int     10h         ; dh – строка, dl – столбец
inc     dh
mov     dl, 8
cursor  dh, dl      ; перевод курсора на следующую строку
mov     cx, razmern
mov     bp, 0       ; bp – смещение по строке
mov     si, 0       ; si – смещение по столбцу
cikl1:  push cx      ; цикл по строкам матрицы
mov     cx, razmern
cikl2:  call read    ; цикл ввода строки матрицы
mov     matrbp[si], al
inc     si          ; переход к след. элементу строки матрицы
add     dl, 4
cursor  dh, dl      ; перевод курсора для ввода следующего
loop    cikl2       ; элемента строки
add     bp, si       ; переход к следующей строке матрицы
mov     si, 0
inc     dh
mov     dl, 8
cursor  dh, dl      ; перевод курсора на следующую строку
pop     cx
loop    cikl1
pop     bp
pop     dx
pop     si
pop     cx
pop     bx
pop     ax
endm

```

Оглавление

I. Информатика	3
<i>Лабораторная работа № 1.....</i>	<i>3</i>
<i>Лабораторная работа № 2.....</i>	<i>5</i>
<i>Лабораторная работа № 3.....</i>	<i>7</i>
<i>Лабораторная работа № 4.....</i>	<i>10</i>
<i>Лабораторная работа № 5.....</i>	<i>12</i>
<i>Лабораторная работа № 6.....</i>	<i>16</i>
II. Языки программирования	24
<i>Лабораторная работа № 1.</i>	<i>24</i>
<i>Лабораторная работа № 2.....</i>	<i>27</i>
<i>Лабораторная работа № 3.....</i>	<i>27</i>
<i>Лабораторная работа № 4.....</i>	<i>31</i>
Приложение 1. Команды ассемблера	35
Приложение 2. Структура программы	43
Приложение 3. Пример макробιβотеки.....	44

Учебное издание

И н ф о р м а т и к а

Лабораторный практикум

Составители: **Власова** Ольга Владимировна
Полякова Ольга Павловна

Редактор, корректор А.А. Аладьева
Компьютерный набор, О.В. Власова, О.П. Полякова
Компьютерная верстка И.Н. Ивановой

Подписано в печать 17.05.05. Формат 60х84/16. Бумага тип.
Усл. печ. л. 2,79. Уч.-изд. л. 1,5. Тираж 50 экз. Заказ


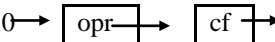
Оригинал-макет подготовлен
в редакционно-издательском отделе ЯРГУ
Отпечатано на ризографе
Ярославский государственный университет.
150000 Ярославль, ул. Советская, 14.

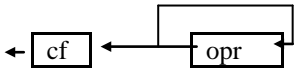
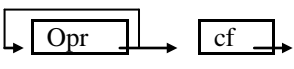
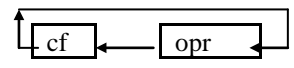
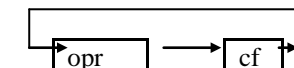
Наименование операции Мнемокод	Содержание операции	Операнды	Код команды	Изменяемые флаги
Пересылка MOV dst,src	dst ← src	R/M,REG	1000 10 D W MOD REG R/M	Не меняются
		НЕПОСР → R/M	1100 011W MOD 000 R/M D8 D16	
		НЕПОСР → REG	1011 W REG D8 D16	
		ПАМЯТЬ → AX/AL	1010 00 D` W adrm adrs	
		R/M,SEG	1000 11 D 0 MOD 0 SEG R/M	
Положить в стек PUSH org		R/M	1111 1111 MOD 110 R/M	Не меняются
		REG	0101 0 REG	
		SEG	000 SEG 110	
Взять из стека POP org		R/M	1000 1111 MOD 110 R/M	Не меняются
		REG	0101 1 REG	
		SEG	000 SEG 111	
Флаги в стек PUSHF			1001 1100	Не меняются
Флаги из стека POPF			1001 1101	Не меняются
Регистры в стек PUSHA			0110 0000	Не меняются
Регистры из стека PUSHA			0110 0001	Не меняются
Перестановка XCHG dst,src	Dst ← src src ← dst	REG,R/M	1000 011 W MOD REG R/M	Не меняются
		REG,AX/AL	1001 0 REG	
Пересыл. адреса	dst ← адр. scr	REG16 ← ПАМЯТЬ	1000 1101 MOD REG R/M	Не меняются

LEA dst,src				
Загр. сегм. данн. LDS dst,src	$dst \leftarrow \text{адр. scr}$ $DS \leftarrow \text{SEG scr}$	REG16,ПАМЯТЬ32	1100 0101 MOD REG R/M	Не меняются
Загр. сегм. ES. LES dst,src	$dst \leftarrow \text{адр. scr}$ $ES \leftarrow \text{SEG scr}$	REG16,ПАМЯТЬ32	1100 0100 MOD REG R/M	Не меняются
Сложение ADD dst,src	$dst \leftarrow dst+src$	REG,R/M	0000 00 D W MOD REG R/M	SF ZF CF OF PF AF
		R/M+НЕПОСР. ОП.	1000 00 S W MOD 000 R/M D8 D16	
		AX/AL+НЕПОСР.ОП	0000 010 W D8 D16	
Сложение с переносом ADC dst,src	$dst \leftarrow dst+src+CF$	REG,R/M	0001 00 D W MOD REG R/M	SF ZF CF OF PF AF
		R/M+НЕПОСР. ОП	1000 00 S W MOD 010 R/M D8 D16	
		AX/AL+НЕПОСР.ОП	0001 010 W D8 D16	
Увеличение на 1 INC opr	$opr \leftarrow opr+1$	R/M	1111 111 W MOD 000 R/M	SF ZF CF OF PF AF
		REG	0100 0 REG	
Вычитание SUB dst,src	$dst \leftarrow dst-src$	REG,R/M	0010 10 D W MOD REG R/M	SF ZF CF OF PF AF
		R/M-НЕПОСР. ОП	1000 00 S W MOD 101 R/M D8 D16	
		AX/AL- НЕПОСР.ОП	0010 110 W D8 D16	
Вычитание с переносом SBB dst,src	$dst \leftarrow dst-src-CF$	REG,R/M	0001 10 D W MOD REG R/M	SF ZF CF OF PF AF
		R/M-НЕПОСР. ОП	1000 00 S W MOD 011 R/M D8 D16	
		AX/AL- НЕПОСР.ОП	0001 110 W D8 D16	
Декремент DEC opr	$opr \leftarrow opr-1$	R/M	1111 111 W MOD 001 R/M	SF ZF CF OF PF AF
		REG	0100 1 REG	

Изменение знака NEG opr	$\text{opr} \leftarrow (-\text{opr})$	R/M	1111 011 W MOD 011 R/M	SF ZF CF OF PF AF
Умнож. беззн. MUL src	$\text{AX} \leftarrow \text{AL} * \text{src}$ $\text{DX}:\text{AX} \leftarrow \text{X} * \text{src}$	R/M	1111 011 W MOD 100 R/M	CF OF
Умнож. знаков. IMUL src	$\text{AX} \leftarrow \text{AL} * \text{src}$ $\text{DX}:\text{AX} \leftarrow \text{AX} * \text{src}$	R/M	1111 011 W MOD 101 R/M	CF OF
Деление беззнаковое DIV src	$\text{AL} \leftarrow \text{AX} \text{ div } \text{src}$ $\text{AH} \leftarrow \text{AX} \text{ mod } \text{src}$ ----- $\text{AX} \leftarrow \text{DX}:\text{AX} \text{ div } \text{src}$ $\text{DX} \leftarrow \text{DX}:\text{AX} \text{ mod } \text{src}$	R/M	1111 011 W MOD 110 R/M	Неопределенное
Деление знак. IDIV src	$\text{AL} \leftarrow \text{AX} \text{ div } \text{src}$ $\text{AH} \leftarrow \text{AX} \text{ mod } \text{src}$ $\text{AX} \leftarrow \text{DX}:\text{AX} \text{ div } \text{src}$ $\text{DX} \leftarrow \text{DX}:\text{AX} \text{ mod } \text{src}$	R/M	1111 011 W MOD 111 R/M	Неопределенное
Сравнение CMP op1,op2	op1-op2	REG,R/M	0011 10 D W MOD REG R/M	SF ZF CF OF PF AF
		R/M- непосред. оп	1000 00 S W MOD 111 R/M D8 D16	
		AX/AL- непосред. оп	0011 110 W D8 D16	
Преобразование Байт-слово	$\text{AX} \leftarrow \text{AL}$		1001 1000	Неопределенное

CBW				
Преобразование Слово-дв.слово CWD	$DX:AX \leftarrow AX$		1001 1001	Неопределенно
Отрицание NOT opr	$opr \leftarrow \overline{opr}$	R/M	1111 011W MOD 010 R/M	
Конъюнкция AND dst,src	$dst \leftarrow dst \& src$	REG,R/M	0010 00 D W MOD REG R/M	SF ZF PF CF=OF=0
		R/M & непосред. оп	1000 00 S W MOD 100 R/M D8 D16	
		AX/AL & непосред.оп	0010 010 D8 D16	

Дизъюнкция OR dst,src	$\text{dst} \leftarrow \text{dst} \vee \text{src}$	REG,R/M	0000 10 D W MOD REG R/M	SF ZF
		R/M \vee непосред. оп	1000 00 S W MOD 001 R/M D8 D16	PF
		AX/AL \vee непосред. оп	0000 110 W D8 D16	CF=OF ==0
Исключаю- щее или XOR dst,src	$\text{dst} \leftarrow \text{dst} \oplus \text{src}$	REG,R/M	0011 00 d w mod reg r/m	SF ZF
		R/M \oplus непосред. оп	1000 00 S W MOD 110 R/M D8 D16	PF
		AX/AL \oplus непосред. оп	0011 010 W D8 D16	CF=OF ==0
Проверка бит TEST st,src	Dst & src	REG,R/M	1000 010 W MOD REG R/M	SF ZF
		R/M & непосред. оп	1111 011 W MOD 000 R/M D8 D16	PF
		AX/AL & непосред. оп	1010 100 W D8 D16	CF=OF ==0
Лог. сдвиг влево SHL opr,cnt		R/M, непосред. оп или CX	1101 00 V W MOD 100 R/M	SF ZF PF CF
Лог. сдвиг вправо SHR opr,cnt		R/M, непосред. оп или CX	1101 00 V W MOD 101 R/M	SF ZF PF CF
Арифм. Сдвиг влево SAL opr,cnt	$\text{opr} \leftarrow \text{opr} * 2^{\text{cnt}}$	R/M, непосред. оп или CX	1101 00 V W MOD 100 R/M	SF ZF PF CF
Арифм. Сдвиг вправо SAR opr,cnt	$\text{opr} \leftarrow \text{opr} / 2^{\text{cnt}}$	R/M, непосред. оп или CX	1101 00 V W MOD 111 R/M	SF ZF PF CF

Цикл..сдвиг влево ROL opr,cnt		R/M,непоср. оп или CX	1101 00 V W MOD 000 R/M	SF ZF PF CF
Цикл. сдвиг вправо ROR opr,cnt		R/M,непоср. оп или CX	1101 00 V W MOD 001 R/M	SF ZF PF CF
Цикл..сдвиг влево с перенос. RCL opr,cnt		R/M,непоср. оп или CX	1101 00 V W MOD 010 R/M	SF ZF PF CF
Цикл. сдвиг вправо с перен. RCR opr,cnt		R/M,непоср. оп или CX	1101 00 V W MOD 011 R/M	SF ZF PF CF
Безусловный переход JMP адрес		Короткий	1110 1011 адрес	
		Прямой внутрисегмент.	1110 1001 adrm adrs	
		Косвенный внутрисегм.	1111 1111 MOD 100 R/M	
		Прямой межсегментн.	1110 1010 adrm adrs segm segs	
		Косвенный межсегм.	1111 1111 MOD 101 R/M	
Вызов под-программы CALL адрес		Прямой внутрисегмент.	1100 1000 adrm adrs	Не меня ются
		Косвенный внутрисегм.	1111 1111 MOD 010 R/M	
		Прямой межсегментн.	1001 1010 adrm adrs segm segs	
		Косвенный межсегм.	1111 1111 MOD 011 R/M	

цикл	LOOP мет.	$CX \leftarrow CX-1, CX \neq 0$	Короткие переходы от -128 до 127	E2 смещение	
	LOOPE мет. LOOPZ мет.	$CX \leftarrow CX-1,$ $CX \neq 0$ или $ZF=1$		E1 смещение	
	LOOPNE м. LOOPNZ м.	$CX \leftarrow CX-1,$ $CX \neq 0$ или $ZF=0$		E0 смещение	
	JCXZ метка	$CX=0$		E3 смещение	
Условный переход	JE	$op1 = op2 \ ZF=1$	Любые	74 смещение	После команды CMP или команд, изменяющих флаги
	JZ	$ZF=1$			
	JNE	$op1 \neq op2 \ ZF=0$	Любые	78 смещение	
	JNZ	$ZF=0$			
	JL / JNGE	$op1 < op2$	Знаковые	7C смещение	
	JLE / JNG	$op1 \leq op2$	Знаковые	7E смещение	
	JG / JNLE	$op1 > op2$	Знаковые	7D смещение	
	JGE / JNL	$op1 \geq op2$	Знаковые	75 смещение	
	JB / JNAE	$op1 < op2$	Без знака	72 смещение	
	JBE / JNA	$op1 \leq op2$	Без знака	76 смещение	
	JP / JPE	$PF=0$			
	JA / JNBE	$op1 > op2$	Без знака	77 смещение	
	JAЕ / JNB	$op1 \geq op2$	Без знака	73 смещение	
	JO	$OF=1$		7A смещение	
	JS	$SF=1$		70 смещение	
	JNO	$OF=0$		71 смещение	
	JNS	$SF=0$		79 смещение	

Возврат из подпрограммы RET		Далекий	1100 1011	Не меняются
		Близкий	1100 0011	
Возврат из подпрограммы RET n	n- число байтов занимаемые параметрами в стеке	Далекий	1100 1010 D8 D16	Не меняются
		Близкий	1100 0010 D8 D16	
Прерывание INT номер			1100 1101 номер	IF TP
Очисть флаг DF CLD	DF=0		FC	
Установ.флаг DF STD	DF=1		FD	
Очисть флаг CF CLC	CF=0		F8	
Измен. флаг CF CMC	CF=1- CF		F5	
Установ.флаг CF STC	CF=1		F9	

И н ф о р м а т и к а

Лабораторный практикум

