ЗАДАНИЯ НА ЛАБОРАТОРНЫЕ РАБОТЫ ПО АССЕМБЛЕРУ

***Первое задание.***

Требования

----------

Задание выполняется на Visual C++ (любая версия с 2003) использованием

ассемблерных вставок. В этом задании необходимо выполнить соответствующие

преобразования над строкой или строками. Решение задачи необходимо оформить

в виде одной или несколько подпрограмм, содержащих ассемблерные вставки. Как

правило, в каждом задании по одной или двум входным строкам надо

получить выходную строку, удовлетворяющую определенным условиям, причем

под выходную строку необходимо выделить память и сделать это надо внутри

ассемблерной вставки. Кроме того, программа должна иметь "дружелюбный"

интерфейс (например, предлагать выполнить повторное тестирование). Ввод

данных из файла не требуется, хотя приветствуется. Ввод/вывод с

консоли выполнять с помощью функций printf и scanf, вызов которых

тоже должен происходить внутри ассемблерных вставок.

Задачи

------

1. Удалить из исходной строки все пробельные символы, пробегая строку

справа налево.

2. Выделить из строки подстроку указанной длины, начиная с заданной

позиции.

3. Определить номер позиции, с которой начинается первое слева

вхождение указанной строки символов.

4. Создать строку, полученную копированием исходной строки заданное число

раз.

5. Указать те символы, которые есть и в первой и во второй строке.

6. Указать те символы, которые есть в первой строке, но которых нет во

второй.

7. Выделить те символы, которые встречаются в исходной строке

указанное количество раз.

8. Найти в строке символы с минимальным и максимальным кодом.

9. Из исходной строки удалить все слова, встречающиеся более одного раза.

10. Из исходной строки удалить все вхождения заданной

последовательности символов.

11. Найти всех соседей заданного символа в исходной строке. Первый и

последний символы являются соседями.

12. Посчитать количество символов, у которых равные соседи в исходной

строке. Первый и последний символы являются соседями.

13. Удалить из исходной строки все символы, встречающиеся в другой.

14. Получить строку, обратную к заданной, то есть все символы которой

записаны в обратном порядке.

15. Удалить все пробелы в конце строки.

16. Удалить из строки первое вхождение заданной последовательности символов.

17. Переставить в обратном порядке все символы между первым и

последним вхождением заданного символа в исходной строке, если этот символ

встречается не менее двух раз.

18. Вставить после заданного символа все символы предшествующие ему.

Оставшуюся часть строки оставить без изменения.

19. Вставить после заданного символа все символы предшествующие ему в

обратном порядке. Оставшуюся часть строки оставить без изменения.

20. Образовать строку, повторив фрагмент исходной строки с заданной

позиции данной длины требуемое число раз.

21. Образовать строку из исходной, повторив i-ый элемент i раз.

22. В строке указанное количество символов, начиная с заданной

позиции, переписать в конец строки.

23. Удалить из цепочки все вхождения двух идущих подряд заглавных

латинских букв (три и более оставить на месте).

24. Удалить из первой строки все символы, которых нет во второй.

25. Удалить из строки подстроку, заданную другой строкой.

26. Найти максимальное слово-палиндром в строке.

27. Найти самое длинное слово в строке.

28. Найти в строке максимальную последовательность букв.

29. Пусть каждый символ строки --- цифра 0...9. На первом месте может

быть знак -. Получить строку, являющуюся результатом сложения первой и

второй строки. 30. В строке заменить восемь идущих подряд пробелов

символом табуляции.

30. Посчитать количество слов в строке. Слова отделяются друг от

друга группами пробелов и табуляций.

31. Перевести целое число в строку символов.

32. Перевести строку символов в целое число.

33. Определить длину максимальной подстроки в первой строке, которая

не содержит символов из второй строки.

ЗАДАНИЯ НА ЛАБОРАТОРНЫЕ РАБОТЫ ПО АССЕМБЛЕРУ

***Второе задание***

Задачи

------

Матрицы

Задачи этой группы предполагают работу с целочисленными квадратными

матрицами, хранящимися в виде структуры. Способ представления матрицы в

памяти указывается в задании. С помощью ассемблерных вставок необходимо

реализовать функции: создать матрицу; удалить матрицу; прочитать, изменить

элемент, стоящий на пересечении i-той строки и j-го столбца; печать матрицы;

функции, которая описывается в задании.

1. Вычислить определитель матрицы, разложением по произвольной строке,

считая, что матрица хранится в виде одномерного массива.

2. Вычислить определитель матрицы, разложением по произвольному столбцу,

считая, что матрица хранится в виде массивов строк.

3. Написать операцию умножения матрицы, считая, что матрица представлена в

памяти массивом столбцов.

4. Среди строк матрицы определить те строки, которые являются перестановкой

из n символов, где n --- размерность матрицы.

5. Транспонировать матрицу, записав результат на место исходной без

заведения дополнительной матрицы. Матрица представлена как массив строк.

6. Проверить является ли матрица симметрической, если она хранится в виде

линейного массива.

7. Посчитать количество строк матрицы, сумма элементов которых есть

нечетное число, если матрица хранится в виде массива столбцов.

8. Посчитать количество столбцов, все элементы которых меньше по модулю

заданного числа. Матрица представлена массивом строк.

Сортировка и поиск экземпляров структур данных

Задачи этой группы предполагают работу с массивом фиксированной

размерности указателей на структуры определенного типа. С помощью

ассемблерных вставок необходимо написать функции: создание и удаление

экземпляра структуры; печать экземпляра на экране; сравнение двух экземпляров

по критерию, указанному в задании (если есть); операцию, указанную в задании.

9. struct Human //Человек

{

char Name[30]

int year;

};

Сравнение: Сравниваются лексикографически поля Name, затем, в случае

равенства, поля year. Отсортировать пузырьковой сортировкой в порядке

убывания.

10. struct BookCard //Книжная карточка

{

char Title[30];

char Author[30];

};

Написать функцию, которая по двум заданным строкам: имя автора и название

книги (возможно пустым или содержащим любую часть слова) находит все

подходящие карточки.

11. struct BookCard //Книжная карточка

{

char Title[30];

char Author[30];

};

Сравнение: сначала сравниваются поля Author, а затем, в случае равенства,

поля Title. Удалить из массива все повторные вхождения каждой из карточек.

После работы массив не должен содержать между любыми двумя указателями на

карточку указатель, не связанный ни с какой карточкой.

12. struct Human //Человек

{

char Name[30];

int date; //количество дней от 1 января 1900 года с момента

рождения

};

Вывести на экран список людей, у которых дни рождения в указанном месяце.

13. struct Human //Человек

{

char Name[30];

int date;\\количество дней от 1 января 1900 года с момента рождения

};

Вывести на экран тех людей, которые родились в один и тот же день и месяц,

но, возможно, в разные годы.

14. struct BookCard //Книжная карточка

{

char Title[30];

char Author[30];

};

Найти все книги указанного автора и отсортировать их по возрастанию (операция

сравнения та же самая, что и в задаче 11).

15. struct Shape

{

int count; //количество вершин

int\* vertexes; /\*одномерный массив вершин, то есть

координаты каждой вершины содержатся в элементах

массива с индексами

2n и 2n+1\*/

};

Сравнение: сначала сравниваются поля count, а, в случае равенства, вершины в

лексикографическом порядке. Отсортировать фигуры по убыванию.

16. struct date

{

int year;

int month;

int date;

};

Отсортировать массив дат по месяцам, а внутри одного месяца, по году и числу

в порядке убывания.

Длинные числа

Задачи этой группы предполагают работу с целыми длинными числами

представленными в памяти массивом чисел подходящего размера, указатель на

которой хранится в структуре. Написать с помощью ассемблерных вставок

операции: создать длинное число по строке; удалить длинное число; печать не

экран; операция указанная в задании.

17. Сложить и вычесть два целых двоичных длинных числа, если в массиве

хранятся двойные слова.

18. Умножить два целых двоичных длинных числа, если в массиве хранятся

слова.

19. Разделить целое двоичное длинное число на второе, если в массиве

хранятся байты.

20. Сложить и вычесть два целых упакованных двоично-десятичных длинных

числа.

21. Умножить целое неупакованное двоично-десятичное длинное число на другое.

22. Выполнить целочисленное деление неупакованного двоично-десятичного

длинного числа на другое.

ЗАДАНИЯ НА ЛАБОРАТОРНЫЕ РАБОТЫ ПО АССЕМБЛЕРУ

***Третье задание***

Вычисление функции разложением в ряд Тейлора

Вычислить значение функции f, которая указана в задании, в точке x,

используя разложение в ряд Тейлора. Суммирование выполнять до тех пор, пока

общий член ряда не будет меньше, по модулю, заданного параметра epsilon.

Решение оформить в виде функции

double имя\_функции(double x, double epsilon =1e-5);

Тестовая программа должна вычислять значение эталонной функции и

вычислять абсолютную погрешность.

1. cos(x)

2. sin(x)

3. tg(x) (вычислите сумму первых 5 членов ряда)

4. ctg(x) (вычислите сумму первых 5 членов ряда)

5. exp(x)

6. ln(x+1)

7. (1+x)^{1/3}

Интеграл

Вычислить значение определенного интеграла функции на отрезке [a, b],

указанным методом с шагом step. Чтобы вычислить значение функции, кроме

многочленов, можно воспользоваться стандартной библиотекой.

Решение оформить в виде функции

double имя\_метода(double a, double b, double (\*pf)(double),

double step= 1e-2);

8. Метод прямоугольников. В качестве примера, реализовать вычисление

интеграла cos(x), x, exp(x).

9. Метод трапеций. В качестве примера, реализовать вычисление интеграла

sin(x), x, ln x.

Нахождение корней уравнений

Найти корень уравнения f(x)=0 на отрезке [a, b] с погрешностью epsilon,

если известно, что корень на указанном отрезке всего один, методом, указанным

в задании.

Решение оформить в виде функции

double имя\_метода(double a, double b, double (\*pf)(double),

double epsilon = 1e-2);

10. Метод деления отрезка пополам. В качестве примера, решите уравнения

cos(x)=0, ln(x)=0 и exp(x)-1/x=0 на отрезке [a, b].

Аналитическая геометрия

В задачах этого раздела точка --- это структура вида

struct Point

{

double x;

double y;

double z;

};

11. Среди множества точек (заданного массивом) в трехмерном векторном

пространстве, найти пару точек наиболее удаленных друг от друга. Вычисление

расстояния между точками оформить в виде отдельной подпрограммы.

12. Среди множества точек (заданного массивом) указать все тройки без учета

порядка, которые лежат на одной прямой. Определение того, что три точки

лежат на одной прямой, выделить в отдельную подпрограмму.

13. Среди множества точек (заданного массивом) указать все четверки без учета

порядка, которые лежат в одной плоскости. Определение того, что 4 точки

лежат в одной плоскости, выделить в отдельную подпрограмму.

14. Геометрическая фигура задана массивом точек, выполнить поворот этой

фигуры относительно оси Oz на угол alpha.

15. Найти точку, из указанного массива точек, расстояние от которой до

заданной плоскости Ax+By+Cx+D=0 будет наименьшим. Вычисление расстояние от

точки до плоскости выделить в отдельную подпрограмму.

16. Среди множества плоскостей выделить все плоскости, параллельные между

собой. Определение параллельности двух плоскостей оформить в виде

подпрограммы.

Матрицы вещественных чисел и их определители

В задачах этого раздела квадратная матрица --- это структура, в которой

хранится размер матрицы и указатель на область памяти, в которой хранится

матрица.

17. Вычислить определитель матрицы, используя разложение по первой строке,

считая, что матрица хранится в памяти, как массив указателей на строки.

18. Решить систему трех линейных алгебраических уравнений от трех

неизвестных методом обратной матрицы.

19. Решить систему трех линейных алгебраических уравнений от трех

неизвестных по правилам Крамера.

20. Найти столбец матрицы (указать номер), модуль суммы элементов которого

максимален. Вычисление модуля суммы оформить в виде отдельной подпрограммы.

21. Найти строку матрицы, сумма модулей отрицательных элементов которого

минимальна. Вычисление суммы модулей оформить в виде отдельной подпрограммы.