Задачи с решениями доступны на сайте

1. Написать ассемблерную программу, определяющую сумму элементов подстроки:
    a. Расширьте код с 6-го слайда презентации для 3-й недели упражнений.
    б. Зарезервируйте место в памяти для еще двух переменных типа.*длинный-инд1иind2*; суммировать элементы между этими двумя индексами
    в. Используйте индексную адресацию, как на упомянутом 6-м слайде.

2. Написать программу на ассемблере, которая находит минимальный и максимальный элемент последовательности отмеченных чисел:
    a. Зарезервируйте место в памяти еще для двух переменных –*мин*и*Макс,*который будет содержать минимальный и максимальный элемент массива после завершения программы
    б. Используйте косвенную адресацию, как на 7 слайде презентации к 3 неделе упражнений.

3. На языке ассемблера напишите программу, удаляющую пробелы из начала строки:
    a. Расширьте код с 14 слайда презентации для 3 недели упражнений.
    б. Сохраните код для удаления пробела в конце строки, которая уже есть на упомянутом слайде; конечная строка должна быть такой, чтобы не содержать пробелов ни в начале, ни в конце строки
    в. В качестве вспомогательного средства используйте код C с 13-го слайда презентации для 4-й недели занятий.
    д.**Идея**-переместить всю строку на одну позицию влево, пока в начале не будет найден непустой символ

4. Написать ассемблерную программу для сортировки массива 16-битных маркированных значений (в произвольном порядке):
    a. Используйте произвольный тип адресации
    б. Доступное решение сортирует массив в порядке возрастания, используя алгоритм пузырьковой сортировки.

Проблемы без решения доступны на сайте

1. Написать ассемблерную программу для определения среднего арифметического последовательности.
    a. Используйте операцию деления
    б. Работа со значениями произвольного размера
    в. Используйте произвольный тип адресации

2. Напишите ассемблерную программу для сортировки строки символов по алфавиту:
    a. Сортировать в произвольном порядке
    б. Зарезервируйте место в памяти и запишите в него строку произвольного размера; отсортировать символы этой строки
    в. Используйте произвольный тип адресации

3. Напишите программу на ассемблере, определяющую покомпонентную сумму двух строк:
    a. Зарезервируйте пространство памяти для двух массивов одинаковой длины с элементами произвольного размера.
    б. Поместите результат в третий массив памяти.
    в. Если добавлены строки*аиб,*и результат помещается в массив*с*, то для каждого элемента массива*с* это действительно*с*[и] ″=″*а*[и] +*б*[и]
    д. Используйте произвольный тип адресации

4. Напишите ассемблерную программу, которая находит индексы всех минимальных и максимальных элементов массива:
    a. Исходный массив может содержать дубликаты, поэтому по нескольким индексам массива можно найти элементы с максимальным и минимальным значением.

б. Для записи результатов зарезервируйте еще две строки: одну для индексов максимальных элементов, а другую для индексов минимальных элементов.

в. Размеры результирующих массивов могут быть равны размеру исходного массива.

д. Используйте произвольный тип адресации