ЗАДАЧИ ДЛЯ ПЕРВОГО СЕМЕСТРА

Тема 2. Задачи на обработку целочисленного массива.

В следующих задачах требуется написать функцию, получающую в качестве параметров имя массива и его длину. По умолчанию (т.е. если по смыслу задачи не предполагается иное) массив целочисленный. Заполнение массива числами из файла, определение его фактической длины должно выполняться отдельной подпрограммой. Функция main открывает файл, вызывает функцию, заполняющую массив числами из файла, распечатывает его, вызывает функцию, преобразующую массив (или вычисляющую его характеристику) и выводит результат (или преобразованный массив) на экран. В задачах 1-20 можно предполагать, что количество чисел в массиве (как в исходном, так и в преобразованном) не превосходит некоторой фиксированной величины (например, 1000). В остальных задачах память под хранения массива должна выделяться динамически (в точности на то количество элементов, которое записано в файле). Файл нельзя открывать более одного раза, нельзя возвращаться к началу файла. Использовать дополнительные массивы нельзя (если в условии об этом ничего не говорится) Ошибки обработать. Некоторые определения (участок, локальный минимум и т.д.) содержатся в списке задач на обработку последовательности.

1. Функция должна возвращать 1, если массив симметричный и 0 в противном случае.

2. Функция должна так преобразовать массив, чтобы его элементы шли в обратном порядке. Использовать не более чем $n/2$ перемещений элементов (n - длина массива)

3. Функция должна так преобразовать массив, чтобы каждый его элемент, кроме первого и последнего, заменился на полусумму соседей.

4. Функция должна циклически сдвинуть элементы массива на одну позицию вправо используя не более чем n перемещений элементов (n - длина массива).

5. Функция должна циклически сдвинуть элементы массива на одну позицию влево используя не более чем n перемещений элементов (n - длина массива).

6. Функция должна циклически сдвинуть элементы массива на k позиций вправо, используя не более чем n перемещений элементов (n – длина массива).

7. Даны два массива (оба массива нужно считывать из файлов). Функция должна так преобразовать первый массив, чтобы каждый его элемент, равный некоторому элементу второго массива, заменился на 0.

8. Функция должна удалить из массива все отрицательные элементы, а оставшиеся сдвинуть к началу массива (уплотнить с сохранением порядка). Возвращать Функция должна размер преобразованного массива. Использовать не более чем n перемещений элементов (n - длина массива)

9. Функция должна удалить из массива все положительные элементы, а оставшиеся сдвинуть к концу массива (уплотнить с сохранением порядка). Освободившиеся места заполнить нулями. Использовать не более чем n перемещений элементов (n - длина массива)

10. Функция должна продублировать (рядом) каждый отрицательный элемент массива. Использовать не более чем $2n$ перемещений элементов (n - длина массива)

11. Циклически сдвинуть на одну позицию влево максимальные по включению постоянные участки.

12. Для заданного числа $k < n$ поменять начальный участок массива (элементы с номерами, меньшими k) с его конечным участком (остальные). Взаимный порядок элементов в обоих участках измениться не должен (n – длина массива). Использовать не более чем n перемещений элементов (n - длина массива)

13. Сравнить два неупорядоченных целочисленных массива A и B как числовые множества: $A = B$ и $A \subset B$. Повторы элементов в массиве не учитывать (сравнение множеств).

14. Назовем x-отрезком группу подряд идущих элементов массива, каждый из которых равен x. Для заданного числа x заменить элементы каждого x-отрезка на полусумму элементов, прилегающих к этому отрезку справа и слева. Если x-отрезок расположен в начале или конце массива, считать недостающий крайний элемент равным нулю.

15. Пусть в массиве последовательно записаны цифры некоторого длинного десятичного числа. Реализовать функции "прибавление единицы" и "вычитание единицы" из такого числа. Вычитание свести к сложению. (вычитание из 0 определить удобным образом). Размер массива может измениться.

16. Заменить все локальные минимумы в массиве одним элементом, значение которого равно 0. Функция должна возвращать количество элементов в преобразованном массиве.

17. Заменить каждый элемент массива количеством элементов массива с меньшими индексами, имеющими значение, меньше данного элемента, т.е. каждый элемент массива a_i заменить количеством элементов a_j, таких что $j < i$ и $a_j < a_i$.

18. Сравнить два неупорядоченных целочисленных массива A и B как числовые мультимножества: $A = B$ и $A \subset B$. Учитывать повторы элементов в массиве.

19. Удалить из массива наиболее часто встречающееся значение. Если таких значений несколько, то выбрать любое из них. Функция должна возвращать количество элементов в получившемся массиве.

20. Удалить из массива наименее часто встречающееся значение. Если таких значений несколько, то удалить все эти значения. Функция должна возвращать количество элементов в получившемся массиве.

21. В каждом участке строгого возрастания в массиве заменить все элементы на среднее значение участка (рассматриваются участки которые нельзя удлинить, т.е. максимальные по включению).

22. Поменять местами в массиве локальные минимумы с соседними порядковыми номерами.

23. Удалить из массива участки строго возрастания с длиной не более 3 (рассматриваются участки которые нельзя удлинить, т.е. максимальные по включению). Функция должна возвращать количество элементов в получившемся массиве.

24. Пусть элементы массива не убывают. Двоичным поиском определить, на какую позицию можно поставить заданное число x, не нарушив порядок. Функция должна возвращать этот индекс.

25. Функция должна отрицательные элементы массива переместить в начало, а положительные – в конец. Количество перемещений элементов не должно более чем n, где n —- размер массива.

26. Функция должна положительные элементы массива переместить в начало, а отрицательные – в конец. Количество перемещений элементов не должно более чем чем n, где n —- размер массива.

27. Даны 2 массива. Функция должна возвращать количество вхождений одного массива в другой (как участок последовательности, в том числе пересекающихся вхождений).

В задачах 28-31 требуется вычислить значение функции $f_n(a_0, ..., a_{n-1})$, заданной индуктивно. На последовательности, состоящей из одного элемента, она принимает значение, равное этому элементу, $f_1(a) = a$. Формула для вычисления $f_2(a, b)$ известна (выделить ее отдельной функцией). Решение не должно использовать рекурсивные вызовы.

28. Для $n > 2$ $f_n(a_0, \ldots, a_{n-1}) = f_2(a_0, f_{n-1}(a_1, \ldots, a_{n-1}))$.

29. Для $n > 2$ $f_n(a_0, \ldots, a_{n-1}) = f_{n-1}(f_2(a_0, a_1), a_2 \ldots, a_{n-1})$.

30. Для $n > 2$ $f_n(a_0, \ldots, a_{n-1}) = f_{n-1}(f_2(a_0, a_{n-1}), a_1, \ldots, a_{n-2})$.

31. Для $n > 2$ $f_n(a_0, \ldots, a_{n-1}) = f_{n-1}(a_1, \ldots, a_{n-2}, f_2(a_0, a_{n-1}))$.

32. В целочисленном массиве хранится перестановка чисел $1, \ldots, n$. Определить ее четность. Каждый элемент просматривается не более O(1) раз.

33 В целочисленном массиве хранится перестановка чисел $1, \ldots, n$. Не используя дополнительных массивов, найти обратную перестановку.

34. Дано число $m <= n$ и массив $(a_0, \ldots, a_{n-1})$. Для каждого его участка из m элементов найти сумму m стоящих рядом элементов. Число действий порядка $O(n)$. Результат сохранить в исходном массиве.

35. В 2-х массивах хранятся коэффициенты многочленов (от одной переменной, степени могут быть различны). Поместить в третий массив коэффициенты произведения многочленов.

36. Даны два неубывающих массива. Соединить их в третий, тоже неубывающий массив. Число действий – порядка суммы размеров исходных массивов.

37. Даны два неубывающих массива. Пересечь их в третий тоже неубывающий массив. Число действий – порядка суммы размеров исходных массивов.

В задачах 38-53 квадратную матрицу следует расположить как единый одномерный массив (функция, преобразующая массив или вычисляющая некоторую характеристику матрицы получает в качестве одного из аргументов int *matr). Размер квадратной матрицы определяется исходя из количества чисел, содержащихся в файле. Если это невозможно, следует выдать соответствующее сообщение и завершить выполнение задачи. Массив нужно распечатывать на экран в виде квадратной матрицы.

38. Все ли строки матрицы одинаковы?

39. Функция должна возвращать сумму диагональных элементов.

40. Функция должна так преобразовать матрицу, чтобы поменялись местами первый и второй столбцы.

41. Функция должна так преобразовать матрицу, чтобы поменялись местами первая и последняя строки.

42. Функция должна транспонировать исходную матрицу.

43. Функция должна возвращать сумму элементов $a_{i,j}$ таких, что $i > j$.

44. Функция должна возвращать 1, если матрица является симметричной и 0 в противном случае.

45. Функция должна так преобразовать матрицу, чтобы последняя строка заменилась на ее сумму с первой.

46. Функция должна так преобразовать матрицу, чтобы последний столбец заменился на его сумму с первым.

47. Функция должна возвращать номер столбца, в котором содержится наибольшее количество нулей.

48. Функция должна возвращать 1, если строки матрицы являются лексикографически упорядоченными по возрастанию и 0 в противном случае.

49. Функция должна удалить первый столбец матрицы.

50. Функция должна возвращать $\max_i \sum_j a_{i,j}$.

51. Функция должна заменить матрицу A на $A + A^t$, где A^t – транспонированная матрица.

52. Функция должна заменить матрицу A на $A - A^t$, где A^t – транспонированная матрица.

53. Переместить все нулевые строки матрицы вниз (сделать их последними, порядок остальных строк можно менять).

В задачах 54-58 нужно так преобразовать целочисленный массив, чтобы его элементы шли в порядке возрастания. Для проверки использовать библиотечную функцию qsort.

54. Отсортировать массив методом пузырька.

55. Отсортировать массив методом выбора максимального элемента.

56. Отсортировать массив методом простых вставок.

57. Отсортировать массив методом бинарных вставок.

58. Отсортировать массив методом подсчета (диапазон значений элементов считать известным). Использовать дополнительный массив размера диапазона.

59. Вычислить k-порядковую статистику массива. Преобразовать его так, чтобы элементы, имеющие индексы меньшие, чем k, были не больше, чем a_k, а остальные элементы – не меньше, чем a_k.