

Постановка

Написать консольное приложение для работы с одномерными массивами.

Задачи

Последовательность функций –

- 1) Считывание массива из ввода пользователя;
- 2) Вывод элементов массива;
- 3) Найти Max, Min элементы и вывести их значения и номера;
- 4) Считать режим работы приложения для последующих действий - На вход подается символ k . Если k равен 'a', то приложение использует Режим 1, если k равен 'b', то Режим 2;
- 5) Выполнить прямую и обратную сортировку в соответствии с режимом – Режим 1 – использование самостоятельно написанных методов сортировки, Режим 2 – методы класса System.Array;
- 6) Заполнить новый массив четными элементами из исходного;
- 7) Вывести массив;

Формат

ввода

10 1 6 67 0 8 49 38 4

Формат

вывода

10 1 6 67 0 8 49 38 4

67 3

0 4

0 1 4 6 8 10 38 49 67

67 49 38 10 8 6 4 1 0

38 10 8 6 4 0

Пояснения к пунктам

задания:

3. Поиск Max и Min элементов осуществлять **ТОЛЬКО** собственным алгоритмом

4. Обработать случаи, когда вводимый результат не удовлетворяет условию выводом сообщения «Режим не выбран, заканчиваю работу...»

5. Прямая – сортировка по возрастанию, обратная – по убыванию.

Постановка

задание консольное приложение для работы с двумерными массивами.

Последовательность функций –

- 1) Считывание массива из ввода пользователя.
- 2) Вывод элементов массива.
- 3) Найти Max, Min элементы и вывести их значения и номера.
- 4) Считывание второго массива.
- 5) Найти произведение, сумму и разность двух массивов.
- 6) Вывести полученные массивы.

Пояснения к пунктам

задания:

3. Поиск Max и Min элементов осуществлять **ТОЛЬКО** собственным алгоритмом .

5. См. операции с матрицами, в случае недопустимости операции вывести соответствующее сообщение «Операция {n} недопустима!», где {n} – название операции (Multiply, Add, Subtract) .

Формат

Ввода

2 3

7 2 1

2 3

5 1 6

-1 -2 0

Вывода

2 3

7 2 1

7 1 0

0 0 1

10 0 18

-7 -4 0

7 1 9

6 0 1

-3 -1 -3

8 4 1

Постановка

задача консольное приложение для работы со ступенчатыми массивами.

Последовательность функций –

- 1) Считывание массива из ввода пользователя.
- 2) Вывод элементов массива.
- 3) Найти Max, Min элементы и вывести их значения и номера.
- 4) Считать введенный пользователем номер элемента массива.
- 5) Изменить элемент на случайный.
- 6) Вывести полученный массив.

Формат

Ввода

2 0 3

7 2 1 5

Формат

2 0 3

Вывода

1 1

2 0 3

7 6 3 2 1 5

Пояснения к пунктам

задания:

3. И использовать класс Random.

Задача 4 – Эксплуатация. Ввод данных

Постановка

задание предыдущие задания так, чтобы можно было осуществлять ввод как с клавиатуры, так и с помощью считывания файла. Функционал -

- 1) Считать режим работы приложения для последующих действий - На вход подается строка s . Если s равен *'kb'* (*клавиатура*), то приложение использует Режим считывания с клавиатуры, если s равен *'fl'* (*файл*), то Режим считывания из файла.
- 2) Все последующие действия написанного приложения, которые требуют ввод, должны использовать соответствующий режим.

Постановка

задания предыдущие задания так, чтобы осуществлялась проверка вводимых данных, осуществлялась запись действий пользователя. «Соединить» предыдущие задания так, чтобы можно было выбрать вариант функционала (Задания 1-3), сделав это в формате меню. Функционал -

- 1) Если на вход вместо необходимых данных поступила некорректная информация (вместо строки чисел – строка из слов) – нужно вывести информацию об ошибке (что за ошибка, где появилась, причины), при этом используя блок `tryparse`.
- 2) Считать режим работы приложения для последующих действий - На вход подается строка i . Если i равен $'t1'$, то приложение использует функционал первого задания. Если i равен $'t2'$, то приложение использует функционал второго задания. Если i равен $'t3'$, то приложение использует функционал третьего задания.

Постановка

задачи предыдущие задания так, чтобы приложение работало до тех пор, пока пользователь не прервет работу с ним, ввод осуществлялся до тех пор, пока не окажется правильным. Весь функционал должен находиться в статических методах. Функционал -

- 1) Весь функционал предыдущих заданий должен быть в статических методах.
- 2) В методе Main должны находиться только метод вызова меню.
- 3) Все пункты заданий, которые требуют ввода пользователя должны повторяться до тех пор, пока не будет осуществлен правильный ввод.
- 4) Как функционал задания был завершен (Задания 1-3), программа переходит в главное меню.

Формат и порядок сдачи

Форма

Лабораторную работу необходимо представить в виде решения .Net Framework. Для каждого задания должен быть создан отдельный проект (консольное приложение) внутри решения. Название проекта должно быть в формате «номер лабораторной» «номер задания». К моменту сдачи проект должен быть полностью рабочим и соответствовать условиям задания, которые сдаются. Задания должны располагаться в папке решения, название которого соответствует формату «группа» «ФИО» «номер лабораторной».

Порядок

сдачи лабораторной **необходимо защитить** её лично. Вопросы будут по работе приложения, написанному функционалу и общим знаниям тем в лабораторной. Лабораторные защищаются **только по одному**, групповая защита невозможна. Сроки сдачи и даты сдачи будут указаны отдельно.

Оценивание и допуск на экзамен

Оценивани

Защищенные задания 1-3 = Оценка «удовлетворительно»

Защищенные задания 1-4 = Оценка «хорошо»

Защищенные задания 1-6 = Оценка «отлично»

Допуск

Для допуска к экзамену необходимо сдать все лабораторные работы на оценку «удовлетворительно».