1. Строгого деления на подгруппы не будет. Кто готов - приходим, н***о заранее, в накануне занятий, присылайте список тех, кто придет сдавать , чтобы все было оперативно и не было толп студентов в связи со сложной обстановкой . Придерживайтесь количества - максимум 20 человек!!!***Все успеем, если будете работать. Как уже говорила, возможны дополнительные консультации в zoom, заранее присылайте вопросы.

2. К 1-й лабораторной работе блок-схема не нужна, если вычисления представляют линейный процесс.

Если решаются 2 и более самостоятельных задач, есть циклы - блок-схема нужна.

3. В отчете по 2-й лабораторной работе блок-схема детальная нужна, чтобы могли объяснить алгоритм решения

4. В отчете по 3 и 4 лабораторным работам - блок-схемы на уровне функций и взаимодействия между ними.

5. Лабораторные можно приносить на флэшке и сдавать на компьютере одногруппника или на кафедральном.

6. Задания на 2-ю и последующие лабораторные берете из методички в соответствие номеру в списке группы.

Кому заданий не хватило, в ближайшее время добавлю в методичку задания и вышлю в поток.

7. Сдавать лабораторные приходите с отчетом, который содержит титульный лист с сайта кафедры, задание, блок-схему (если нужна), код, скриншоты работы. Приносите код на флэшке или приходите с ноутбуком.

8. Сдача лабораторных проходит так:

- демонстрируете работу кода;

- проверяющий может попросить вас не только объяснить, как это работает, но  попросить :

1) что-то оптимизировать или  добавить в код;

2) заменить один цикл на другой;

3) и т.д.

**Не забываем, постоянно повторяю о правильном оформлении кода**

1. Программа должна быть отформатирована, используйте табуляцию или кнопку форматирования, ее структура должна быть понятна.

2. Используйте длинные наглядные имена, например, NumberOfIterations (Number\_Of\_Iterations), LineLength, Number\_Columns, отражающие суть и назначение  переменной

3. Комментарии должны пояснять блоки кода. Когда будете использовать функции - комментируйте их назначение. Недопустимы «лишние», ненужные комментарии, например, «Складываем два числа»

4.  Ваши программы должны быть оптимальны по выделяемой памяти под переменные и времени работы. Обращайте внимание на выбор структур данных. Если задачу можно решить без массива, то его использовать не надо, ограничьтесь простыми переменными. Если вы работаете с файлом, не нужно копировать все данные в массив и далее работать с массивом. Работаете с файлом!

 5. Интерфейс пользователя должен быть «дружественным», т.е. при запуске программы на выполнение пользователю было понятно, что делать. Результат работы должен быть представлен в понятном виде, не забывайте о форматировании при выводе данных на экран. Результаты работы программы должны, по возможности, оставаться на одном экране. Не следует неоправданно часто использовать функцию очистки экрана!!!

6. Программу нужно протестировать дома на тестах, содержащих граничные условия. Программа должна работать на всем множестве допустимых входных данных.

7. Добавить контроль правильности ввода данных, в том числе используя обработку исключений (деление на 0, ввод отрицательных или символьных значений в задаче подсчета возраста человека, оценок за сессию и т.д.). Кто знает, как это сделать - делитесь знаниями в группах.