1) Вывести числа от 1 до N (int N = 100)

2) Переделать первое так, чтобы N бралось из консоли

3) Переделать 2 так, чтобы выводились только четные числа

4) Программа просит у пользователя число. Вывести все его делители

5\*) Вывести все простые числа от 1 до N

6\*) Оставить в 4 только простые делители

7. Выбрать максимум из двух чисел.

8. Выбрать максимум из трех чисел.

9. Выбрать максимум из четырех чисел.

10-13. На доске есть пешка и какая-нибудь фигура (конь, слон, **ладья**, ферзь) по их координатам надо ответить, можно ли съесть пешку на этом ходу.

14. Определить по трем сторонам треугольника – а может ли вообще такой существовать и его вид (тупо прямо остро угольный правильный равнобедренный)

15. Високосный год или нет.

* год, номер которого кратен 400, — високосный
* остальные годы, номер которых кратен 100, — невисокосные (например, го­ды 1700, 1800, 1900, 2100)
* остальные годы, номер которых кратен 4, — високосные

16. На вход подается две даты - вычислить количество дней между ними.

17\*\* - учесть юлианский и григорианский календари

18. И уже классика – квадратное уравнение

19. FIZZBUZZ

/\*

Вывести числа от 1 до 100, заменяя кратные 3 на FIZZ, кратные 5 на BUZZ, а кратные и 3, и 5 - на FIZZBUZZ.

\*/

20-21. счастливые билеты (6 знаков)

/\*

Посчитать количество шестизначных счастливых билетов (сумма первых трех цифр равна сумме последних трех цифр)

\*/

(через 6 циклов и через 1, вытаскивая из счетчика цифры)

22\*\* - обобщить на случай n - значного счастливого билета

Вывести несколько последовательностей – вывод их от 1 до 100, к примеру

23. 1 2 4 7 11 16 и тд

24. 1 4 9 16 25 и тд

25. 1 2 4 5 10 11 22 и тд

26. Фибоначчи – куда без них 1 1 2 3 5 8 13 и тд, 2 2 4 6 10 16 26 и тд a b a+b a+2b и тд

27. “елочка” (звездочки)

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

И тд..

28. Модификация звездочек

1

2 3

4 5 6

7 8 9 10

И тд.

Ну и перед/после имеет смысл повыводить кучу квадратиков (или на дз)

29.

1 2 3 4 5

1 2 3 4 5

1 2 3 4 5 и тд

\_\_\_\_\_\_\_

30.

1 2 3 4 5

6 7 8 9 10

11 12 13 14 15и тд

\_\_\_\_\_\_\_

31.

0 1 .. N

N … 1 0

…

M раз

\_\_\_\_\_\_\_

32.

0 1 2 .. N

0 … N-1

…

0

(мод2 елочка наоборот)

\_\_\_\_\_\_\_

33.

0 1 2 … N

1 2 3 … N+1

…

N … 2\*N

\_\_\_\_\_\_\_

34\*.

0 1 2 3 4

15 16 17 18 5

14 23 24 19 6

13 22 21 20 7

12 11 10 9 8

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

35. Функции арифметических операций.

36. Функция побитовой конъюнкции/дизъюнкции

37. Функция вывода N числа фибоначчи

38. Функция факториала

39. Функция степени

40. Функция экспоненты

41. Функция sin

42. Функция cos

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

43-46. Начальные задачи на массивы - заполнить числами от 1 до 100 + взять все последовательности из первого семинара для заполнения массива.

/\*

1 2 3 4 5 .. 100

2 4 6 .. 200

1 3 5 .. 199

1 2 4 8 ..

\*/

47. Далее можно уже приступать непосредственно к. Минимум/максимум в массиве.

48. Среднее арифметическое+геометрическое.

49. Количество каки-нибудь пар, где правое меньше левого.

50. Склеивание двух массивов.

51\*. Склеивание двух массивов как-нибудь хитро, например, чтобы окончательный был по возрастанию (немного извращений а-ля поиск макс/мин в обоих массивах и впихивание этого в новый массив БЕЗ СОРТИРОВКИ

52. Сдвиг массива (1 2 3 4 5 -> 5 1 2 3 4) Сдвиг массива на n, естественно.

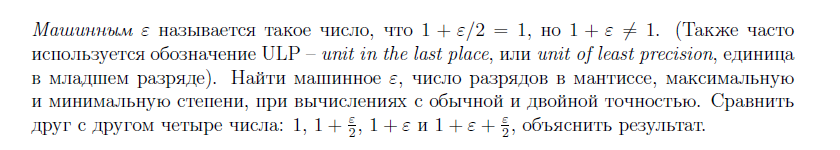
53. переворот (1 2 3 4 5 -> 5 4 3 2 1)

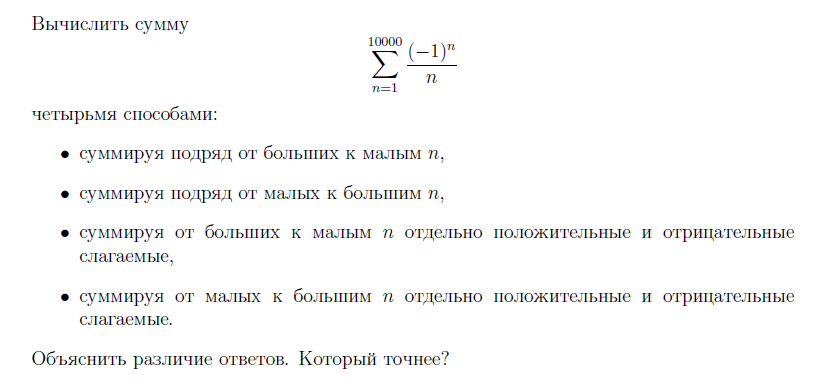
54-56. Квадратичные сортировки - пузырек, вставкой, выбором.

57. После сортировок можно медиану заставить найти.

58. Решето Эратосфена.

59.



60.

61. Переполнение int + UB продолжение следует… (оно появится)

62. swap

63. Игры с указателями - изменить int \* через char \*.

//123456789

/\*

#include <stdio.h>

int main(void) {

int \*a = (int \*) malloc(sizeof(int));

char \*b = a;

//\*b = 123456789;

\*(b) = ;

\*(b+1) =;

\*(b+2) =;

\*(b+3) =;

printf("%d", \*a);

return 0;

}

\*/

64. C(n, k) - с текстовыми файлами

65. Структуры - сортировка массива структур.

66. Односвязные списки - реализовать интерфейс.

/\*

struct List \*addElement(struct List \*head, int value, int typeOfAdding); //or 3(2) functions

struct List \*deleteElement(struct List \*head, int value);

void printList(struct List \*head);

void freeList(struct List \*head);

struct List \*sortList(struct List \*head); // сортировка слиянием

int isCycled(struct List \*head);

\*/

67. Двусвязные списки.

68. Бинарный поиск

69. Топологическая сортировка графа.

70. Строки

ASCII

‘\0’

/\*

size\_t strlen(const char \*);

char \*strcpy(char \*toHere, const char \*fromHere);

intstrcmp(const char \*, const char \*);

char \*strcat(char \*dest, const char \*src);

char \*strstr(const char \*haystack, const char \*needle);

\*/

typedef, enum - на вашей совести.

Сравнение эффективности операций - тоже на вашей совести. На алгоритмах вас должны будут этим хорошо помучить.

Линейный поиск - о чем тут говорить

Двоичные деревья - урезаем в пользу нормального завершения курса.