Задачи на функции:

1. **Фаренгейты в цельсии:** На вход подаются температуры плавления и кипения в Фаренгейтах некого вещества. Вам нужно узнать будет ли оно жидкостью при комнатной температуре (25 градусов цельсия) и напечатать температуры плавления и кипения в цельсиях.
   Формула для перевода Цельсий в Фаренгейты: $T_c = \frac{5}{9}(T_f - 32)$. Была написана следующую программу:

   ```
   #include <stdio.h>
   int main()
   {
           float Tf1, Tf2;
           scanf("%f%f", &Tf1, &Tf2);
           float Tc_room = 25;

           if (Tc_room > 5.0/9.0*(Tf1-32) && Tc_room < 5.0/9.0*(Tf2-32))
                   printf("YES\n");
           else
                   printf("NO\n");

           printf("Temepratures in Celsius %f %f\n", 5.0/9.0*(Tf1-32), 5.0/9.0*(Tf2-32));
   }
   ```

   Видоизмените программу, добавив функцию `float ftoc(float tf)`, которая будет принимать на вход температуру в фаренгейтах и возвращать её в значение в цельсиях.

2. **Максимум:** У вас есть функция max, которая принимает 2 числа и возвращает их максимум:

   ```
   #include <stdio.h>
   int max(int a, int b)
   {
           if (a > b)
                   return a;
           else
                   return b;
   }
   int main()
   {
           int a, b, c;
           scanf("%d%d%d", &a, &b, &c);
           printf("Max of b and c = %d\n", << Ваш код >>);
           printf("Max of a, b, c = %d\n", << Ваш код >>);
   }
   ```

   Используя эту функцию, найти максимум 2-х чисел и максимум 3-х чисел.

3. **Функция, которая ничего не делает:** Написать функцию, которая ничего не принимает, не возвращает и вообще ничего не делает. Используйте `void`. Зачем нужна такая функция?

4. **Функция, которая только возвращает:** Написать функцию **int the_answer()**, которая ничего не принимает, но возвращает целое число равное 42. Вызвать эту функцию в функции **main()**, чтобы вывести на экран строку “The answer is 42”.

5. **Функция, которая только печатает:** Написать функцию **void hello()**, которая ничего не принимает и не возвращает, но выводит на экран строку “Hello”. Вызвать эту функцию в функции **main()** 100 раз в цикле.

6. **Counter iterative:** Написать функцию `void counter_iterative(int n)`, которая принимает число n, ничего не возвращает и печатает числа от 0 до n. Вызвать эту функцию в функции `main()`.

7. **Doubler:** Написать функцию doubler, которая принимает число и возвращает это число, умноженное на 2. Вызвать эту функцию из функции main:

```
// Тут нужно написать функцию doubler
int main()
{
        int a;
        scanf("%d", &a);
        printf("%d\n", doubler(a));
}
```

8. **Doubler with printf:** Написать функцию doublerp, которая принимает число и печатает это число, умноженное на 2. Функция ничего не должна возвращать(void). Вызвать эту функцию из функции main 100 раз в цикле для чисел от 1 до 100.

9. **Doubler with printf and scanf:** Написать функцию doublerps, которая сама считывает число и печатает это число, умноженное на 2. Функция ничего не должна принимать м возвращать(void). Вызвать эту функцию из функции main 100 раз в цикле. Внимание! Эта и следующая задачи, возможно, единственные задачи из всех задач курса в которой вам нужно использовать scanf внутри функции, отличной от main. **Не используйте scanf в внутри своих функций!**

10. **Функция, которая принимает, возвращает, считывает и печатает:** Напишите функцию `float combiner(float a)`, которая принимает вещественное число a, считывает вещественное число b с помощью scanf, печатает их среднее арифметическое($(a+b)/2$) и возвращает их среднее геометрическое ($\sqrt{ab}$). Используйте эту функцию в функции main, чтобы найти среднее арифметическое и среднее геометрическое чисел 42 и 256.
Для нахождения корня вещественного числа вам понадобится функция **sqrt** из библиотеки math.h. Чтобы подключить эту библиотеку вам нужно добавить в исходный код соответствующую директиву include и добавить опцию компилятора **-lm**.
$ gcc -lm <имя исходного файла>

11. **Print time:** Написать функцию `void print_time(int h, int m)`, которая принимает на вход два целых числа (часы и минуты), и выводит строку вида **hh:mm**. Для печати времени в таком формате нужно использовать функцию printf с опцией “%02i:%02i”. Вызвать эту функцию в функции `main()`.

12. **Is prime?:** Написать функцию **int is_prime(int n)**, которая будет проверять является ли число n простым и возвращать 1 если число n простое либо 0 если число n не является простым.

13. **Print primes:** Написать функцию **void print_primes(int a, int b)**, которая будет печатать все простые числа из отрезка [a, b]. Используйте функцию `is_prime` из предыдущей задачи! Вызвать эту функцию в функции **main()**.

Задачи на функции. Рекурсия:

14. **Counter:** Написана рекурсивная функция **void counter(int n)**, которая печатает числа от n до 1.

```
void counter(int n)
{
        if (n > 0)
        {
                printf("%d ", n);
                counter(n-1);
        }
}
int main()
{
        counter(42);
}
```

Видоизмените эту функцию так, чтобы она печатала числа от 1 до n. (то есть в нормальном порядке).

15. **Sum recursive:** Напишите функцию**int sumrec(int n)**, которая рекурсивно вычисляет сумму первых n натуральных чисел. Вызовите эту функцию из main.

Задачи на функции. Функции и массивы:

На вход подаётся натуральное число n и n целых чисел. считайте этот массив в функции main:

```
int main()
{
        int n;
        int arr[1000];
        scanf("%d", &n);
        for (int i = 0; i < n; ++i)
                scanf("%d", &arr[i]);
}
```

16. **Array print:** Напишите функцию `void print_array(int n, int arr[])`, которая принимает на вход массив и печатает все его числа через пробел. В конце строки напечатайте перенос на новую строку.

17. **Array sum:** Напишите функцию `int sum_array(int n, int arr[])`, которая принимает на вход массив и возвращает сумму массива. Вызовите эту функцию в main(), чтобы найти сумму массива. Сама функция не должна ничего печатать.

18. **Array max:** Напишите функцию `int max_array(int n, int arr[])`, которая принимает на вход массив и возвращает максимальный элемент массива. Вызовите эту функцию в main(), чтобы найти сумму массива. Сама функция не должна ничего печатать.

19. **Double array:** Напишите функцию `void double_array(int n, int arr[])`, которая увеличивает каждый элемент массива в 2 раза. Вызовите эту функцию в main(). Напечатайте массив до и после увеличения в 2 раза, используя функцию print_array.

20. **Sort array:** Напишите функцию `void sort_array(int n, int arr[])`, которая сортирует массив методом выбора. Вызовите эту функцию в main(). Напечатайте массив до и после сортировки, используя print_array.

Задачи на функции. Передача по ссылке (дополнительно):

21. **Modify:** Написать функцию **void doublerm(int*)**, которая удваивает число, поступающее на вход, используя указатель на эту переменную. Используйте эту функцию, чтобы вывести на экран кубы чисел от 1 до 100.

22. **Swap:** Написать функцию **swap**, которая меняет значения 2-х переменных типа int местами. Используйте эту функцию в функции **main()**.