1. **Препроцессор** — это текстовый процессор, управляющий текстом файла исходного кода в ходе первого этапа трансляции.
2. #include, #define, #undef, #if, #else, #elif, #endif, #ifdef, #ifndef, #error, #warning
3. Статическая **память** – хранит глобальные переменные и константы;

Стек – хранит локальные переменные, аргументы функций и промежуточные значения вычислений

Куча – динамически распределяемая **память**;

1. **Пространство** **имен** **это** **декларативная** **область,** **в** **рамках** **которой** **определяются** **различные** **идентификаторы** **(имена** **типов,** **функций,** **переменных,** **и** **т.** **д.)**.
2. **Исключение** — это событие при выполнении программы, которое приводит к её ненормальному или неправильному поведению.
3. –
4. **Необработанное** **исключение** остается для обработки вызывающей **функцией**. Это продолжается до тех пор, пока **исключение** не будет обработано. Это происходит с **вызовом** **функции** или без него. Другими словами, если вызывается **функция**, которая не находится в блоке try, **исключение**, которое происходит в этой **функции**, автоматически передается в **стек** **вызовов**.
5. **Статическая** **память** — это область **памяти**, выделяемая при запуске программы до вызова функции main из свободной оперативной **памяти** для размещения глобальных и **статических** объектов, а также объектов, определённых в пространствах имён.
6. В **куче** же вы сами намеренно выделяете **память**(операторы new delete). Следовательно и контроль за освобождением **памяти** лежит на вас.
7. – Если оператор **throw** указан где-то в функции, которая расположена вне **try**, а в нем вызывается эта функция, то происходит раскручивание стека. Как я поняла, так и юзается память из стека. Хз, у нас там были такие функции, они в другом файле лежат, но я не уверена что это про это.
8. Как тут расписать? Показываешь ему директивы в основном спп, как они работают расписано в 5 лк, там легко, копировать сюда долго, почитаешь 1 раз
9. Условная компиляция применяется там, где используются директивы условной компиляции: #ifdef, #ifndef, #if, defined() . Там чекнуть, где в коде и показать ему
10. Тупой вопрос, это в коде просто показать где namespace
11. Блоки try и catch работают вместе. Блок try обнаруживает любые исключения, которые были выброшены в нем, и направляет их в соответствующий блок catch для обработки. Блок try должен иметь, по крайней мере, один блок catch, который находится сразу же за ним, но также может иметь и несколько блоков catch, размещенных последовательно (друг за другом).

Как только исключение было поймано блоком try и направлено в блок catch для обработки, оно считается обработанным (после выполнения кода блока catch), и выполнение программы возобновляется. В языке C++ **оператор throw** используется для сигнализирования о возникновении исключения или ошибки. Для использования оператора throw применяется ключевое слово throw, а за ним указывается значение любого типа данных, которое вы хотите задействовать, чтобы сигнализировать об ошибке