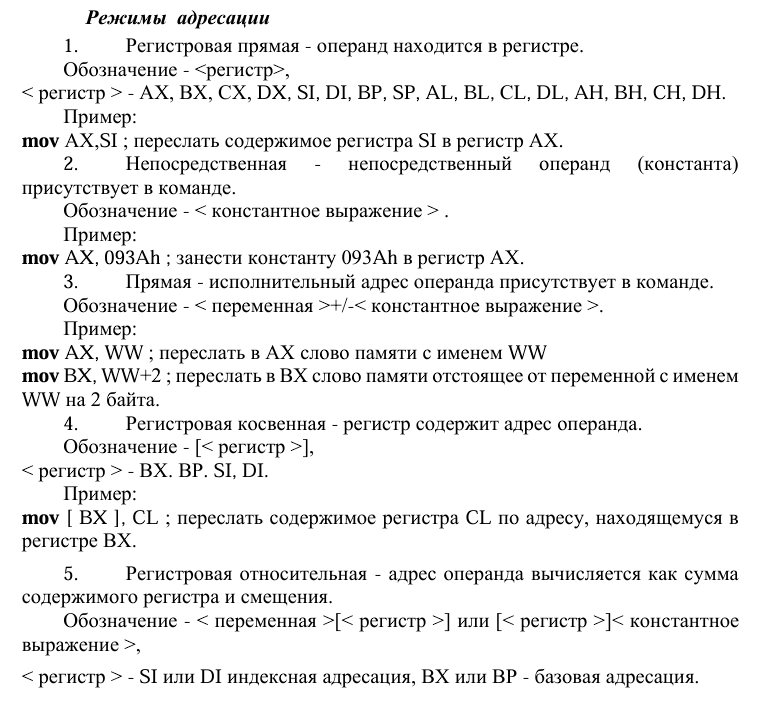
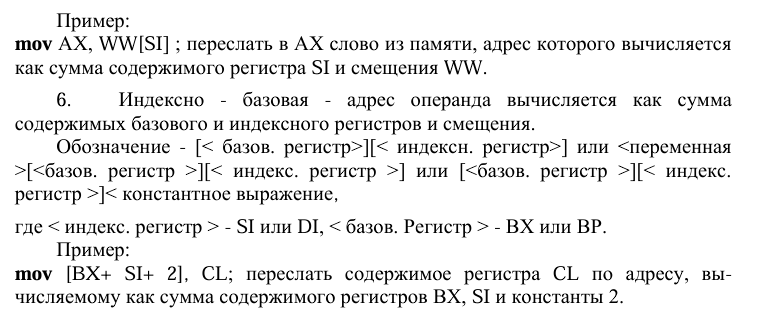
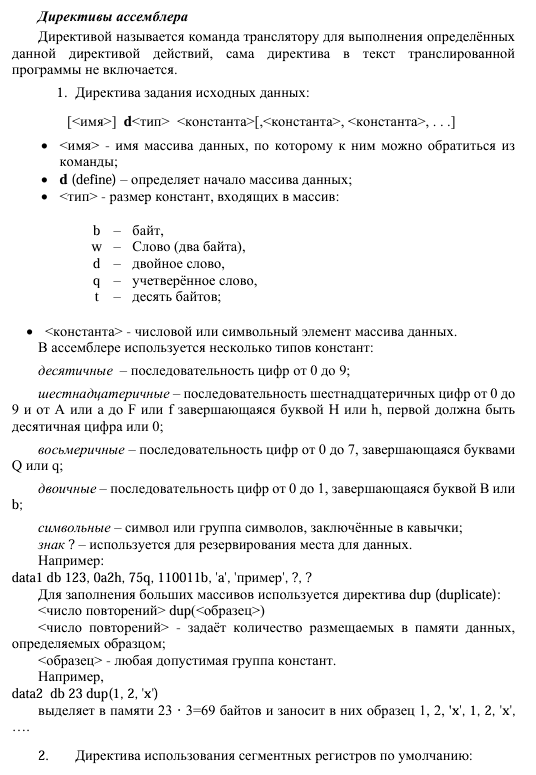
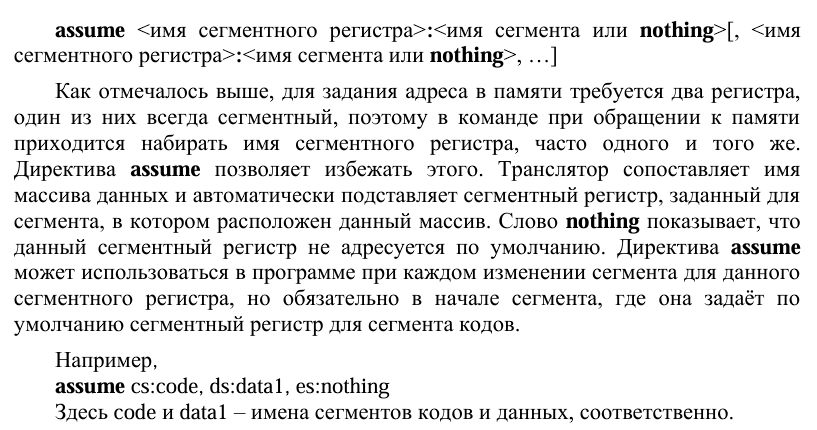
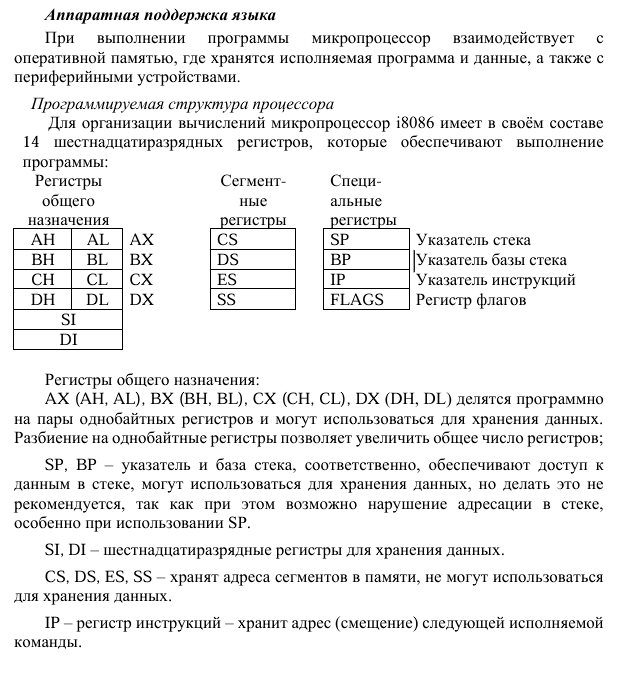
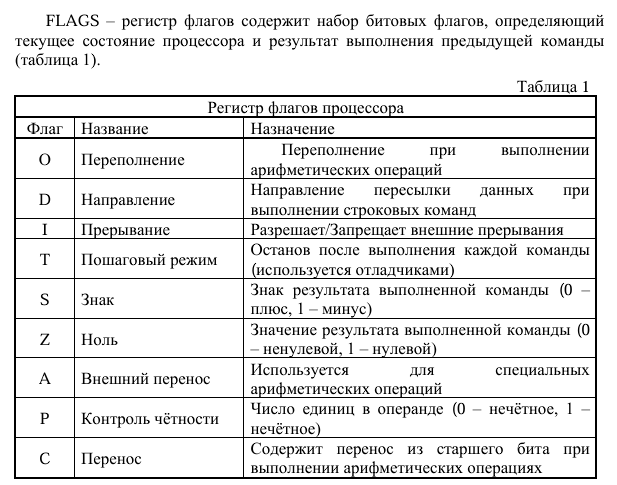
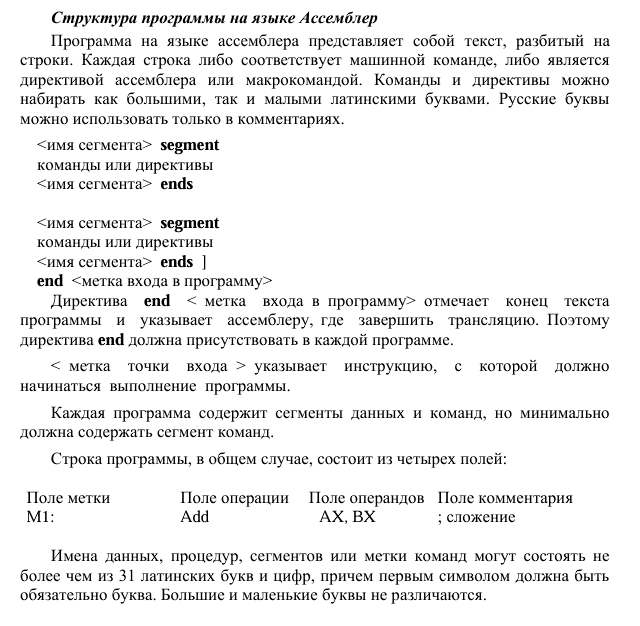
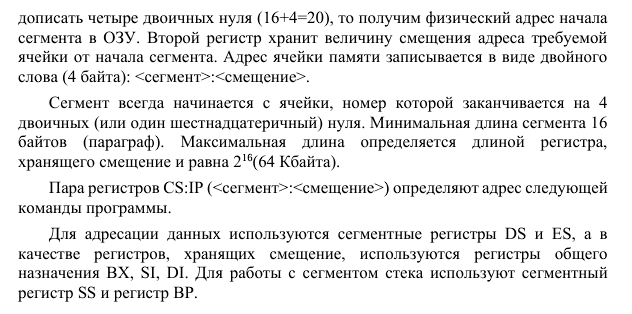
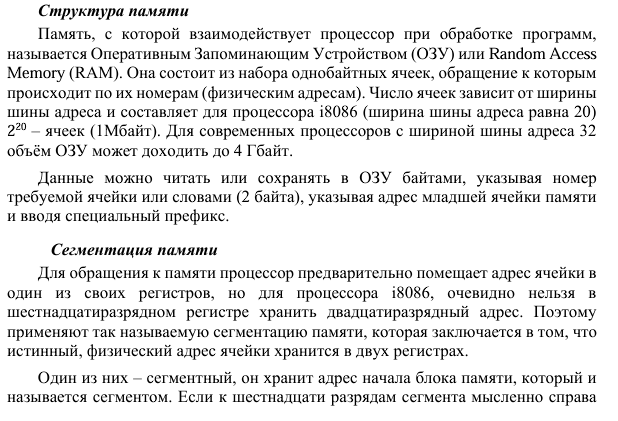
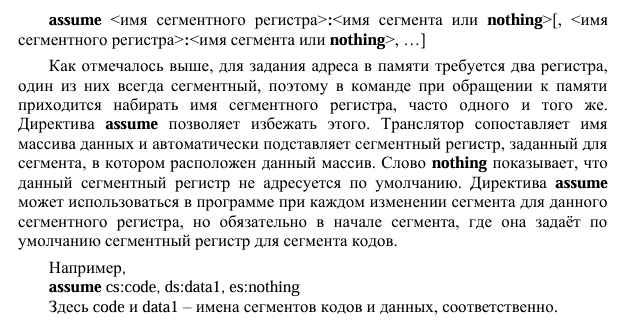
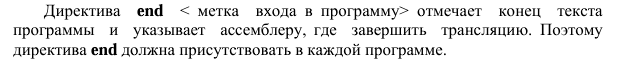
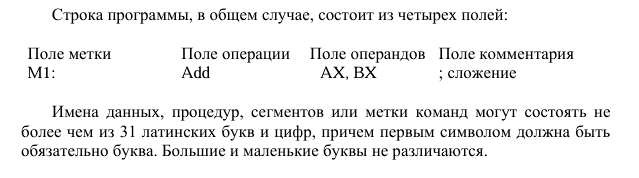
1. Какие режимы адресации присутствуют в языке ассемблер?

1. Что такое директива, назовите доступные директивы в языке ассемблера?  
     
   
2. Поясните структуру регистров процессора  
   
3. Регистр флагов это  
   
4. Структура программы на ассемблере  
   
5. Структура памяти, сегментная память.  
   
6. Назначение директив Segment и Ends.  
   Segment  
   Директива SEGMENT задает имя сегмента. Имя сегмента обязательно должно быть задано, соответствовать соглашениям для имен в Ассемблере и быть уникальным.

Ends  
Директива Ассемблера ENDS указывает на конец описания сегмента. Директивы, SEGMENT и ENDS, должны обладать одинаковыми именами.

1. Назначение дирeктивы assume.  
   
2. Назначение директив DB и DW.  
   создание в структуре данных переменных с размерами в байт (8 бит) и слово (16 бит) соответственно. Они же равны AL, BX, соответственно
3. Назначение оператора DUP в директивах DB и DW.  
   Дублирует данные по следующему паттерну:  
   <число повторений> dub(<образец>)
4. Назначение директивы END.  
   
5. Из каких полей состоит строка программы на Ассемблера?  
   
6. Какие обязательные поля, какие необязательные.  
   Обязательный: поле операции  
   Необязательный: поле комментария, поле операнда (в редких случаях), поле метки (не использовали в коде => не обязателен)
7. В чем различие между командами mov ax, bx ,move ax,[bx], move [ax],bx?  
   mov ax, bx – поместить в AX значение BX  
   mov ax, [bx] – поместить в AX значение из данных, которые находятся в памяти по адресу, равному BX  
   mov [ax], bx – поместить в память по адресу, равному AX, значение BX
8. В чем разница между командой mov a1 и директивой adw1?  
   Mov a, 1 – режим адресации непосредственная  
   A dw 1 – директива задания исходных данных