**1) Ans: There are four segment registers in 8086 microprocessor.**

* **CS** - points at the segment containing the current program.
* **DS** - generally points at segment where variables are defined.
* **ES** - extra segment register, it's up to a coder to define its usage.
* **SS** - points at the segment containing the stack.

although it is possible to store any data in the segment registers, this is never a good idea because the segment registers have a very special purpose - pointing at accessible blocks of memory.  
  
segment registers work together with general purpose register to access any memory value. For example if we would like to access memory at the physical address **12345h** (hexadecimal), we should set the **DS = 1230h** and **SI = 0045h**.

2) ANS:

General purpose registers are used to store temporary data within the microprocessor. There are 8 general purpose registers in 8086 microprocessor.

1. **AX –** This is the accumulator. It is of 16 bits and is divided into two 8-bit registers AH and AL to also perform 8-bit instructions.  
   It is generally used for arithmetical and logical instructions but in 8086 microprocessor it is not mandatory to have accumulator as the destination operand.

Example:

ADD AX, AX (AX = AX + AX)

1. **BX –** This is the base register. It is of 16 bits and is divided into two 8-bit registers BH and BL to also perform 8-bit instructions.  
   It is used to store the value of the offset.

Example:

MOV BL, [500] (BL = 500H)

1. **CX –** This is the counter register. It is of 16 bits and is divided into two 8-bit registers CH and CL to also perform 8-bit instructions.  
   It is used in looping and rotation.

Example:

MOV CX, 0005LOOP

1. **DX –** This is the data register. It is of 16 bits and is divided into two 8-bit registers DH and DL to also perform 8-bit instructions.  
   It is used in multiplication an input/output port addressing.

Example:

MUL BX (DX, AX = AX \* BX)

1. **SP –** This is the stack pointer. It is of 16 bits.  
   It points to the topmost item of the stack. If the stack is empty the stack pointer will be (FFFE)H. It’s offset address relative to stack segment.
2. **BP –** This is the base pointer. It is of 16 bits.  
   It is primary used in accessing parameters passed by the stack. It’s offset address relative to stack segment.
3. **SI –** This is the source index register. It is of 16 bits.  
   It is used in the pointer addressing of data and as a source in some string related operations. It’s offset is relative to data segment.
4. **DI –** This is the destination index register. It is of 16 bits.  
   It is used in the pointer addressing of data and as a destination in some string related operations. It’s offset is relative to extra segment.