Q.1 Output after every Stage of compiling : -

1. Preprocessing : The first stage of compilation is called preprocessing. In this stage, lines starting with a # character are interpreted by the preprocessor as preprocessor commands. These commands form a simple macro language with its own syntax and semantics. This language is used to reduce repetition in source code by providing functionality to inline files, define macros, and to conditionally omit code.

Compiler option used is /Fa, output is .i file

1. Compiling : In this stage, the preprocessed code is translated to assembly instructions specific to the target processor architecture. These form an intermediate human readable language.

Compiler option used is /Fa, output is .asm ifle

1. Assembly : During this stage, an assembler is used to translate the assembly instructions to object code. The output consists of actual instructions to be run by the target processor.

Compiler option used is /Fo, output is .obj file

1. Linking : The object code generated in the assembly stage is composed of machine instructions that the processor understands but some pieces of the program are out of order or missing. To produce an executable program, the existing pieces must be rearranged and the missing ones filled in. This process is called linking.

Compiler option used is /Fe, output is .exe file

Q. 2

1. Output : Compile error
   * This code will result in compiler error
   * This is mainly due to syntax error. During compilation the max(10,20) will be replaced in the code by ((10>20)?10:20); Since ";" marks the end of the line but there is additional "<<endl;" which will give syntax error;
2. Output :Compile error
   * The PRINT macro after if will be replaced by printf("Star Wars"); printf("Psycho");
   * So there will be an else without an if block
3. Output: square of 4: undefined behaviour
   * The square(x++) is replaced by x++\*x++;
   * For some compiler : The value of x=5 so first x++ will increment the value by x so x will become 6 but since x++ is a post increment operator the value assigned during multiplication will be 5\*x++ but since the value of x now is 6 it will be assigned to second x++ that is, the expression becomes 5\*6.Now the value of x becomes 7 but the output becomes 30.
   * For some compiler : The whole function is fed value 5, i.e. both values of x will be assigned with 5, so answer will come 25.
4. d)Output: 55
   * Namespace allows to group class, objects, function, variable under same name. Here the name of the namespace is space. Any variable within a namespace can be accessed by namespace name followed by scope resolution operator followed by variable name. In this code the value of the variable in the space namespace is set to 5. So, the output is 55
5. Output: Compilation Error
   * truefalse (assuming there is a type error, second: is actually second::x or second::y, for both cases result is false)