1. Explain the functions of the following terms as detail as possible: (25%)
   (a) Bootstrap loader;(b) Assembler;(c) Handler;(d) File allocation table;(e) RPC

2. A *complier* is used to translate a source program created by high-level programming language into its corresponding object codes for executing by a computer.
   (a) Give the basic scheme of a simple *one-pass* complier and explain the operations of the scheme. (10%)
   (b) How does an interpreter work? (5%)
   (c) What is a complier-complier? (5%)
   (d) What is a *bytecode* (P-code) complier? (5%)

3. A *macro instruction* is simply a notational convenience for the programmer to represent a commonly used group of statements in the source programming language.
   (a) Give the fundamental functions that are common to all macro processors. (10%)
   (b) How should a programmer decide whether to use a macro or a subroutine to accomplish a given logical function? Give an example. (10%)
   (c) What is a keyword macro parameter? (5%)

4. An *operating system* is a program that used to manage the computer hardware. It also provides a basis for application programs and acts as an intermediary between the computer user and the computer hardware.
   (a) List the services provided by an operating system that are designed to make it more convenient for *users* to use the computer system. (5%)
   (b) A *monitor* type is a fundamental high-level synchronization construct used to solve the Dining-Philosophers problem. Describe how it works. (10%)
   (c) *Paging* is a memory-management scheme that permits the physical address space of a process to be noncontiguous. *Segmentation* is also a memory management scheme that supports the separation of the user's view of memory and the actual physical memory. Explain why sharing a reentrant module is easier when segmentation is used than when pure paging is used. (10%)