



**Part I. True/False questions** (Only accept the answer in writing **T** for True or **F** for False. No other answer writings are allowed. 1% is for each question, and **10% for the total of this part.**)

1. Different control sections are most often used for subroutines or other logical subdivisions of an assembly program.
2. A multi-pass assembler is an extension to the two-pass design logic that allows an assembler to handle *forward references* during symbol definition.
3. Dynamic linking is also called load on call.
4. Allowing several executing programs to share one copy of a subroutine or library (instead of linking a separate copy into each object program: linkage editor) is one of the advantages of dynamic linking.
5. To do the three parts of processing in the compilation process – scanning, parsing, and code generation – it is important that a compiler must make three passes over the program being translated.
6. The task of scanning the source statement, recognizing and classifying the various tokens, is known as *lexical analysis*.
7. Infrared signals cannot be used in wireless data communication.
8. In SIMD, there are multiple control units, which can broadcast program instructions to multiple ALUs.
9. Code generation is the work of the second phase in most of compilers.
10. Code optimization can truly make an inefficient algorithm efficient.

**Part II. Multiple Choice** (In the following questions, the answer may have only one or more choices. 5% is for each question, and **25% for the total of this part.**)

1. Which of the following is correct?
  - a). A grammar is the entire collection of rules for a language.
  - b). A grammar that allows the construction of two or more distinct parse trees for the same statement is said to be unambiguous.
  - c). The grammar does describe the *semantics*, or *meaning*, of the various statements; such knowledge may be also supplied in the *code-generation routines*.
  - d). If there is only one parse tree for a given statement, the grammar is said to be unambiguous.
  - e). The code-generation technique may involve a set of routines, *one for each rule* or *alternative rule* in the grammar.
2. Which of the following is incorrect?
  - a). A computer network is a set of independent computer systems connected by telecommunication links and its purpose is to share information and resources.
  - b). For local area networks (LAN), the owner of the devices is also the owner of the means of communications.
  - c). Wide area networks (WAN) connect devices that are across town, across the country, or across the ocean, and users must purchase telecommunications services from an external provider.



- d). Protocols govern the exchange of binary digits across a physical communication channel are data link protocols.
- e). Assigning port numbers to programs and remembering which program goes with which port, is a part of the transport layer protocols.
3. Which of the following is correct?
- In procedural languages, the programmer can devise parallel the sequence of imperative commands in order to speed up the execution of computers.
  - C is originally designed for systems programming and also used for general-purpose computing.
  - In a functional programming language, primitive functions are part of the language and other functions can be defined and named by the programmer. Once defined, functions can be used in the definition of other functions.
  - The Java programmer compiles source code just once in his/her machine. Therefore, Java bytecode interpreter does not need to present on each user machine.
  - SQL is the language used to frame knowledge base queries.
4. Which of the following is incorrect?
- Loading is to bring the object program into memory for execution.
  - A linkage editor always performs linking after loading.
  - Dynamic linking performs all linking and relocation at load time.
  - A bootstrap loader is different from an absolute loader, which is not permanently resident in a read-only memory (ROM).
  - If a program is to be executed many times without being reassembled, the use of linking loader is preferred.
5. Which of the following is correct?
- Control sections can be assembled together or assembled independently.
  - The assembler accomplishes the logical rearrangement of code by maintaining, during Pass 1 and 2, two location counters for all top program blocks.
  - The separation of an assembly program into a few program blocks is considerably reduced the addressing problems in translating source codes to object codes.
  - The load-and-go assembler is a type of assembler that produces object code directly in memory for immediate execution.
  - A multi-pass assembler can make as many passes as are needed to process the definitions of symbols.

**Part III. Other Questions (65% is for the total of this part.)**

(10%) 1. Consider the following program segment, where  $i, j$ , and  $k$  are integer variables.

**for  $i := 1$  to 20 do**



```

for j := 1 to i do
  for k := 1 to j do
    print (i * j + k)
  
```

How many times is the **print** statement executed in this program segment?

(5%) 2. Let  $m \in \mathbb{Z}^+$  with  $m$  odd. Prove that there exists a positive integer  $n$  such that  $m$  divides  $2^n + 1$ .

3. (a)(5%) Construct the number sequence as a binary search tree. The number sequence is: 8, 3, 10, 1, 6, 4, 7, 13. (b)(5%) What is the traversal sequence by using Deep-First-Search algorithm (DFS)?

(10%) 4. A DRAM chip uses two-dimensional address multiplexing. It has 11 common address pins, with the row address having one bit more than the column address. And it has 16 data pins.  
 (a) What is the capacity of this memory chip? (b) How many of the chips are needed to provide a memory capacity of 32Mx32?

(10%) 5. (a) Construct a AB flip-flop, using a D flip-flop and some gates. AB functions are listed as below: 00(complement), 01(set), 10 (clear) and 11(no change). Show the characteristic equation and its circuit. (b) Complete following AB excitation table.

Q	Q(t+1)	A	B
0	0		
0	1		
1	0		
1	1		

(10%) 6. What are the conditions that lead to a deadlock?

(5%) 7. Consider the following code:

```

#include "stdio.h"
#include "stdlib.h"
int my_recursive_function(int a, int b)
{
  if (a>b)
  {
    if ((a%b)!=0)
      return my_recursive_function(b, a%b);
    else
      return b;
  }
}
  
```



```
    }
    else
    {
        if ((b%a)!=0)
            return my_recursive_function(a, b%a);
        else
            return a;
    }
}
int main()
{
    printf("%d \n",my_recursive_function(247, 1463));
    return 0;
}
```

(5%) 8. What will happen when the following code is executed?

```
int my_recursive_function(int n)
{
    If (n <=1)
        return n;
    else
        return n+my_recursive_function(n-2);
}
int main()
{
    printf("%d", my_recursive_function(10));
    return 0;
}
```

- a) 55
- b) 30
- c) 20
- d) 10