NEW

2015

BCA

3rd Semester Examination

SYSTEM PROGRAMMING

PAPER—2102

Full Marks : 70

Time : 3 Hours

The figures in the right-hand margin indicate full marks.

Candidates are required to give their answers in their own words as far as practicable.

Illustrate the answers wherever necessary.

Answer *Question no. 1* and any *four* from the rest.

1. Answer any *five* questions :　　　　　　　5×2

   (a) What is a system program ? How does it differ from application program ?

   (b) What is self assembler ?

   (c) What do you mean by forward references ?

   (d) Write down the benefits of high level language over assembly language.

   (e) What is 'Boot-Strap' loader?

   (f) What is preprocessor? How does it differ from macroprocessor?

   (g) Compare macro and subroutine.

2. (a) What is linking? How does it different from loading?

   (b) Give an example to explain linking of external references.

   (c) What is the function of a two-pass direct linking loader?

$$(2+3)+5+5$$

3. (a) Define an assembler with an example. Write step-by-step 'what happened when computer first turn on or restarted'.

   (b) Draw and explain the flow chart of pass-2 of two pass assembler. Also describe the data structures used in pass-2.

   (c) Why do you need load-and-go assembler?

$$(2+3)+7+3$$

○

         (Continued)

4. (a) What do you mean by backward reference ?

(b) Write an assembly language program to add two numbers using macro. (The addition operation is performed within macro and values are supplied as arguments).

(c) Explain nesting of macro with suitable example.

2+8+5

5. (a) What will happen when you perform static linking with shared libraries ?

(b) Explain the working principle of 'dynamic binding'.

(c) What is recursive macro ?

(d) Explain the drawbacks of absolute loader.

3+7+2+3

6. (a) What is the difference between directive statement and indirective statement ?   3

(b) Explain the following terms :   4×3

(i) Macro parameter ;

(ii) Recursive macro call ;

(iii) Macro definition table ;

(iv) Macro processor.