**Sem. - I**

**System Programming**

**CHAPTER -1**

**Introduction to System Programming**

**1. Write a SMACO program to print CGD of two integers.(Oct.2006)(5 m)**

**2. Write a SMACO program to check whether given number is perfect or not.**

**(April 2007)(5 m)**

**CHAPTER -2**

**Editors**

**Questions for 1 mark.**

1. **List different types of editors. (Oct.2004)**
2. **What is structure editor? (Oct. 2008)**
3. **Which editors do not display text in the manner it would appear if printed. (April 2008)**
4. **Give examples of structure editor. (April 2009)**
5. **With the help of suitable diagram, explain the designing of an editor. (April 2009)**
6. **Which editors maintain the multiple representation of text? (Oct. 2009)**
7. **What is software tool? Give two examples? (April 2007)**

**Questions for 5 marks**

**1. Explain design of editor with the help of suitable diagram.**

**(April 2005 & 2007)**

**2. Write a note on editor. (Oct.2005)**

**CHAPTER -3**

**Assembler**

**Question for 1 mark**

**1. List the assembler directive statements you are familiar with. (Oct. 04)**

**2. What is the purpose of 200 in start 200 statement. (April 05)**

**3. What will be the value of the location counter (LC) at the line number 3 & 4**

**1. START 100**

**2. A DS 3**

**3. b dc ‘1’**

**4. STOP (April 09)**

**4. List the name of data structure of assembler. (Oct. 04, April 06)**

**5. What is literal? Give the general form of literal declaration. (April 09)**

**6. List different kinds of statements used in assembly language. (April 05, 07)**

**7. Define forward reference. (April 2008)**

**8. List the assembly mnemonic in which first operand is not used. (Oct. 2009)**

**9. What is use of symbol table? (Oct. 2009)**

**Question for 5 marks**

**1. What is forward reference? How it is handled by assembler.**

**(April 05, 06, 07)**

**2. Explain EQU and LTORG statements giving examples. (Oct. 04)**

**3. What is purpose of LTORG statement? (April 07)**

**4. describe various tasks performed by single pass assemble. (April 05)**

**5. Give name description of the data structures used for two pass assembler.**

**Explain with suitable examples, the entries in different data structures.**

**(April 06)**

**6. Give the syntax of EQU and use of EQU statement. (April 06)**

**7. What are the different factors affecting pass structure of an assembler. (April 06)**

**8. Compare and Contrast variant I and variant II of intermediate code for**

**Assembly language program. (April 06)**

**9. Generate the intermediate code variant I and II for the following program:**

**START 300**

**READ A**

**READ B**

**MOVER AREG, A**

**ADD AREG, B**

**MOVEM AREG, C**

**MULT AREG = ‘4`**

**SUB AREG, X**

**COMP AREG, ZERO**

**LTORG**

**PRINT C**

**STOP**

**A DS 100**

**B DS 2**

**ZERO DS ‘0`**

**C DS 3**

**X DS 2**

**END (April 2009)**

**10. Write an assembly language program to find out maximum of two numbers. (April 2009)**

**11. What is forward reference? How it is handled by single pass? (Oct. 2009)**

**12. Discuss the various issues related to program listing and error reporting**

**of an assembler. (April 08, Oct. 09)**

**13. What is the role of assembler directives in assembly program? How are**

**They processed? (Oct. 2009)**

**14. List the static data structures required for assemblers. (April 2008)**

**15. Consider the following assembly language program.**

**START 200**

**A DS 1 200**

**LI READ A 201**

**READ B 202**

**MOVER AREG, A 203**

**SUB AREG, B 204**

**MOVEM AREG, C 205**

**PRINT C 206**

**STOP 207**

**B DS 2 208**

**C DS 1 210**

**Show the content of data structures at the end of one pass assembly**

**Scheme.**

**CHAPTER -4**

**Macros and Macro Processors**

**Question for 1 mark**

**1. Give syntax and use of AIF statement. (Oct. 2004)**

**2. Define Macro. (Oct. 2005)**

**3. Values of expansion time variables (EU) can be manipulated through which**

**statement? Write syntax and explain that statement. (Oct. 2004)**

**4. Explain AIF in short. (April 2005)**

**5. Give syntax and use of AGO statement. (Oct. 2005)**

**6. What is positional parameter? (April 2005)**

**7. What is macro call? Give syntax. (April 2007)**

**8. Give syntax for macro call statements. (Oct. 2006)**

**9. Give schematic of macro processor. (April 06)**

**10. What is nested macro call? Give syntax. (April 07)**

**11. Give syntax and use of AIF statements. (April 07)**

**12. Define Macro processor. (Oct. 08)**

**13. What are the contents of macro definition? (Oct. 07)**

**14. Explain keyword parameters and position parameters with suitable**

**example. (Oct. 09)**

**15. Define lexical expansion of a macro. (April 08)**

**16. Give the general form of defining local and global expansion time**

**variables in macro. (April 09)**

**17. Explain the data structures which are shared by all macro defined in a**

**single program. (Oct. 09)**

**18. State the role of sequence symbol. (April 09)**

**19. What are the contents of macro definition? (Oct. 09)**

**20. Keyword parameter can preceded the positional parameters? Justify with**

**Suitable examples. (April 09)**

**21. Define macro assembler. (April 09)**

**22. What is the purpose of preprocessor statement? (Oct. 09)**

**Question for 3 mark**

**1. Explain keyword parameters? (April 05)**

**2. Explain schematic of macroprocessor with help of suitable diagram.**

**(April 05)**

**Question for 6 mark**

**1. Give the names of description of data structures used for macro. Explain**

**with suitable examples the entries in different data structure. (Oct. 05)**

**Question for 5 mark**

**1. Explain keyword parameters and positional parameters with suitable**

**Examples. (Oct. 05)**

**2. Explain how nested macro calls are handled. (Oct. 04)**

**3. Macro definition contains which types of statements? (Oct. 05)**

**4. What is macro processor? Explain design overview of it. (April 07)**

**5. What is macro? Differentiate between formal parameter with actual**

**parameter, give examples. (April 06)**

**6. Explain the data structures which are shared by all macro defined in a**

**single program. (Oct. 07)**

**7. What do you mean by a macro with mixed parameter and with nested**

**macro call? Give an example of each. (Oct. 06)**

**8. Write short note on designing of macro assembler. (Oct. 08)**

**9. Explain keyword parameters and positional parameter with suitable**

**example. (Oct. 07)**

**10. Explain the preprocessing of nested macro with suitable example.**

**(Oct. 08)**

**11. Explain the all tasks involved in macro expansion. (April 08)**

**12. Consider the following macro:**

**MACRO**

**CLEAR MEM \*X, &N, &REG =AREG**

**LCL &M**

**&M SET 0**

**MOVER & REG, = ‘0`**

**.MORE MOVEM &REG, &X + &M**

**&M SET &M + 1**

**AIF (&M NE N). MORE**

**MEND**

**Show the contents of the following data structures :**

**1) MDT 2) SSTAB 3) KPDTAB (April 08)**

**CHAPTER - 5**

**Compilers**

**Question for 1 mark**

**1. Define basic block. (April 05, Oct. 05, April 09)**

**2. What is dynamic pointer? (Oct. 04)**

**3. What is a block in a programming language? (April 08)**

**4. Define dynamic binding. (Oct. 05)**

**5. Define data type. (April 05, 08)**

**6. Define scope of a program entity. (Oct. 06)**

**7. Apply the elimination of common subexpression technique on:**

**a = b \* c;**

**x = b \* c + 5; (April 08)**

**8. What is dead code? (April 09)**

**9. Define memory binding. (Oct. 09)**

**Question for 5 mark**

**1. Explain the concept of AR. Explain with example how AR is created for**

**nested blocks. (Oct. 04, 05)**

**2. What is storage allocation in case of compilation of a program? When is it**

**performed? What is static and dynamic storage allocation? (Oct. 05)**

**3. Explain different code optimization techniques. (Oct. 04)**

**4. Construct triple and indirect triple for the following strings.**

**a + b \* c + d \* e ↑ f & x + b \* c**

**(April 05, Oct. 05, April 07, 09, 2010)**

**5. What are the different intermediate code representation for expression?**

**(April 08)**

**6. Define translation time address.**

**7. How memory allocation of array take place:**

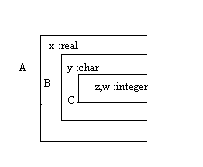
**(a) if dimensions and bounds of an array are known at compile time.**

**(b) if dimensions and bounds of an array are known at compilation time.**

**(Oct. 09)**

**8. For the following block structured program where each block is without**

**Parameter.**



**It is assumed that the blocks are entered in the sequence A,B,C during**

**execution.**

**(a) Show the dynamic allocation using AR when blocks A and B are active.**

**(b) Show the dynamic allocation using AR when blocks A and B are active**

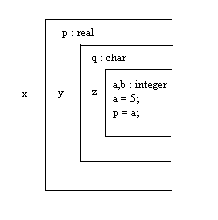
**And block C is entered.**

**(c) What are the actions taken at the entry of block C?**

**(d) What are the actions taken when block C is exited?**

**9. For the following block structured program, where each block could be a**

**procedure without parameters. (April 08)**



**It is assumed that the blocks are entered in the sequence x, y, z during**

**execution.**

**(a) Show the dynamic allocation using AR when blocks x and y are active.**

**(b) Show the dynamic allocation using AR when blocks x and y are active**

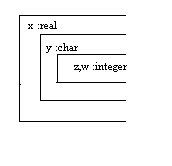
**And block z is entered.**

**(c) What are the actions taken at the entry of block z?**

**(d) What are the actions taken when block z is exited?**

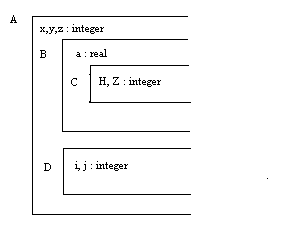
**10. What is Display? Show the contents while executing the program state-**

**ment given below: (Oct. 06)**



**11. Construct the local and non-local variable scope or accessibility table for**

**the following blocks A, B, C, D. (Oct. 08)**



**CHAPTER - 6**

**Compiler Design Options**

**Question for 1 mark**

**1. What is P-code compiler? (April 08)**

**Question for 5 mark**

**1. Write a note on P-code compiler. (Oct. 04, April 05)**

**CHAPTER – 7**

**Linker and Loader**

**Question for 1 mark**

**1. Define self-relocating program. (April 06)**

**2. Define memory binding. (Oct. 04)**

**3. What is link origin? (Oct. 05)**

**4. What is address sensitive instructions? (April 05)**

**5. What is translated origin? (Oct.04)**

**6. Define linking. (Oct. 05)**

**7. Define translation time address. (Oct. 08, 09)**

**8. What do you mean by program location? (Oct. 06)**

**9. Define memory binding. (Oct. 09)**

**10. Differentiate between EXTRN and ENTRY. (April 08)**

**11. The size of relocatable program is larger than non-relocatable program.**

**Comment. (April 07)**

**12. Draw the schematic of program execution. (Oct. 09)**

**Question for 5 mark**

**1. There would be no need for linkers if all programs are coded as self-**

**relocating programs. Comment. (Oct. 05)**

**2. What is program relocation? When it is performed? Explain necessity of it.**

**(Oct. 04)**

**3. Self-relocating programs are less efficient than relocatable programs.**

**Comment. (April 05)**

**4. Write a short note on overlays. (Oct.08, 09)**

**5. What is loader? What are the different types of loader? Define linking.**

**(Oct. 08, 09)**

**6. Write a note on program relocatibility. (Oct.06, April 07)**

**7. What are the different functions performed by the loader? (April 06)**

**8. Write a short note on relocatable program and self relocatable program.**

**(April 09)**

**9. Explain the contains of object module of a program, which contains all**

**information necessary to relocate and link the program with other**

**programs. (April 08)**