

BACHELOR OF SCIENCE IN COMPUTER SCIENCE

First Year Examination – Semester I

SCS 1301 – Data Structures and Program Design in C

Sample Paper B

Two (02) Hours

Answer All Questions

Instructions to Candidates

1. The medium of instruction and questions is English.
2. Note that questions appear on both sides of the paper. If a page or part of this question paper is not printed, please inform the supervisor immediately.
3. This paper has two (02) parts across 12 pages.
4. MCQ should be marked on the given MCQ answer sheet and the essay question should be answered on the answer book provided.
5. Write your index number clearly on all answer sheets.
6. This paper has two (02) parts, Parts I and II. Part I consists of thirty-five (35) MCQ questions for seventy (70) marks, and Part II consists of essay questions for thirty marks (30).
7. Calculators, Mobile Phones, Smart Watches or any electronic devices capable of storing and retrieving data are not allowed.
8. All rights reserved.

******* Multiple Choice Questions Marking Method *******

- Each MCQ question has five (05) answers with one (01) or more (up to 05) correct answers. Each MCQ question is worth two (02) marks. Suppose a question has two (02) correct answers and three (03) incorrect answers, +2 marks will be shared between the correct options (1 mark per option), and –2 marks will be shared between the incorrect options (–0.67) per option. If one correct and incorrect answer are marked, the final mark for the question will be $1 + (-0.67) = 0.33$.
- Each question can have a positive or a negative value as the final mark. Both negative and positive values would be added together to calculate the mark for the MCQ section. If a student scores less than zero (0) for the MCQ section (Part I), zero (0) would be considered as the mark for that section (Part I).

This is a practice paper written to match the structure, question types and difficulty of the SCS 1301 past papers. It is not an official University of Colombo examination paper.

PART I

1. Which of the following statement(s) is/are true regarding second (2nd) generation programming languages?
 - (a) They contain human-readable mnemonics such as MOV and ADD.
 - (b) They are converted to machine code using an assembler.
 - (c) They are machine independent.
 - (d) Their instructions map essentially one-to-one onto machine instructions.
 - (e) They are translated using an interpreter.

2. Which of the following statement(s) is/are true regarding interpreted and hybrid language implementations?
 - (a) An interpreter translates and executes the source statement by statement.
 - (b) Java is compiled to bytecode which is then executed by a virtual machine.
 - (c) Interpreted languages generally execute faster than compiled languages.
 - (d) Python and JavaScript are commonly interpreted.
 - (e) A hybrid implementation produces a machine-code executable ahead of time.

3. Which of the following is/are valid identifiers in C?
 - (a) `_buffer`
 - (b) `2ndValue`
 - (c) `node_count`
 - (d) `double`
 - (e) `Queue1`

4. Which of the following statement(s) is/are true regarding `const` and `typedef`?
 - (a) A `const` object has a type, occupies storage and has an address.
 - (b) `#define` is processed by the compiler, not the preprocessor.
 - (c) `typedef` creates an alias for an existing type, not a new type.
 - (d) A `#define` directive must be terminated by a semicolon.
 - (e) `const` declarations must be initialised at the point of declaration.

Consider the following header file “`defs.h`” and the associated source file “`test.c`” to answer the Questions 5 to 9.

----- `defs.h` -----

```
1 #include <stdio.h>
2 #define PR(fmt, val) printf(#val " =%" #fmt "\t", (val))
3 #define NL putchar('\n')
4 #define PRINT1(f, x1) PR(f, x1),NL
5 #define PRINT2(f, x1, x2) PR(f, x1),PRINT1(f, x2)
6 #define PRINT3(f, x1, x2, x3) PR(f, x1),PRINT2(f, x2, x3)
7 #define PRINT4(f, x1, x2, x3, x4) PR(f, x1),PRINT3(f, x2, x3, x4)
```

----- test.c -----

```
1  #include "defs.h"
2
3  int main(){
4      int x, y, z;
5
6      x = y = z = 1;
7      if(x = 0) y = 2; else y = 3;
8      PRINT2(d,x,y);
9
10     x = 2; y = 3;
11     z = x++ + ++y;
12     PRINT3(d,x,y,z);
13
14     x = 1; y = 2; z = 3;
15     if(x < y)
16         if(y > z) z = 10;
17     else z = 20;
18     PRINT3(d,x,y,z);
19
20     x = 0; y = 0;
21     switch(x){
22         case 0: y += 1;
23         case 1: y += 2; break;
24         case 2: y += 4;
25     }
26     PRINT2(d,x,y);
27
28     x = 5;
29     y = (x > 3) ? (x > 4 ? 1 : 2) : 3;
30     PRINT2(d,x,y);
31 }
```

5. Which of the following is/are (a) value(s) printed by the statement in line 8?

- (a) x = 0
- (b) x = 1
- (c) y = 1
- (d) y = 2
- (e) y = 3

6. Which of the following is/are (a) value(s) printed by the statement in line 12?

- (a) x = 2
- (b) x = 3
- (c) y = 4
- (d) z = 5
- (e) z = 6

7. Which of the following is/are (a) value(s) printed by the statement in line 18?

- (a) $y = 2$
- (b) $z = 3$
- (c) $z = 10$
- (d) $z = 20$
- (e) The `else` in line 17 belongs to the `if` in line 15.

8. Which of the following is/are (a) value(s) printed by the statement in line 26?

- (a) $y = 1$
- (b) $y = 2$
- (c) $y = 3$
- (d) $y = 7$
- (e) The case `0`: body falls through into case `1`:.

9. Which of the following is/are (a) value(s) printed by the statement in line 30?

- (a) $x = 5$
- (b) $y = 1$
- (c) $y = 2$
- (d) $y = 3$
- (e) $y = 5$

Consider the following source file and answer the Questions 10 to 14.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int add(int a, int b){ printf("%d", a + b); return a + b; }
5 int sub(int a, int b){ printf("%d", a - b); return a - b; }
6 int mul(int a, int b){ printf("%d", a * b); return a * b; }
7
8 int main(int argc, char *argv[]){
9     int (*ops[])(int,int) = { add, sub, mul };
10    int a = 6, b = 3;
11    for(int i = 0; i <= atoi(argv[1]); i++)
12        (*ops[i])(a + i, b);
13    return 0;
14 }
```

10. Which of the following is/are correct regarding the declaration in line 9?

- (a) `ops` is an array of three pointers to functions taking two `int` and returning `int`.
- (b) `ops` is a pointer to a function returning an array of three `int`.
- (c) `ops[0](1,2)` and `(*ops[0])(1,2)` are equivalent.
- (d) The `&` operator is required when storing a function name in the array.
- (e) A typedef such as `typedef int (*Op)(int,int);` would allow the declaration to be written as `Op ops[] = { add, sub, mul };`.

11. What would be the output if the integer value one (1) is provided as the command line argument?
- (a) No output
 - (b) 9
 - (c) 94
 - (d) 9424
 - (e) 2449
12. What would be the output if the integer value two (2) is provided as the command line argument?
- (a) No output
 - (b) 9
 - (c) 94
 - (d) 9424
 - (e) 2449
13. If the loop in line 11 is changed to `for(int i = atoi(argv[1]); i >= 0; i--)`, what would be the output if the integer value two (2) is provided as the command line argument?
- (a) No output
 - (b) 9424
 - (c) 2449
 - (d) 4249
 - (e) 2494
14. Which of the following is/are correct regarding the program if it is executed with no command line argument at all?
- (a) `argc` would be 1.
 - (b) `argv[1]` would be `NULL`.
 - (c) `atoi(NULL)` is undefined behaviour.
 - (d) The program would safely print nothing and return 0.
 - (e) Checking `argc` before using `argv[1]` would avoid the problem.
15. Consider the loop `int j = 0; for(int i = 1; j <= n; i++) j = j + i;`. Which are accurate asymptotic complexities for the number of iterations?
- (a) $O(\log n)$
 - (b) $O(\sqrt{n})$
 - (c) $O(n)$
 - (d) $\Theta(\sqrt{n})$
 - (e) $\Theta(n)$

16. Which of the following is/are accurate asymptotic complexities for `deleteFromHead()` and `deleteFromTail()` respectively in a **singly** linked list that maintains both head and tail pointers?

- (a) $O(1)$ and $O(1)$
- (b) $O(1)$ and $O(n)$
- (c) $\Theta(1)$ and $\Theta(n)$
- (d) $O(n)$ and $O(n)$
- (e) $\Omega(1)$ and $\Omega(1)$

17. Which of the following is/are (an) algorithm(s) that use the divide and conquer strategy?

- (a) Binary Search
- (b) Bubble Sort
- (c) Merge Sort
- (d) Quick Sort
- (e) Selection Sort

18. Which of the following is/are correct regarding the program given below?

```
#include <stdio.h>
int main() {
    char arr[] = "SCS1301";
    const char *ptr = "SCS1301";
    printf("%zu %zu %zu\n", sizeof arr, sizeof ptr, sizeof "SCS1301");
    return 0;
}
```

- (a) `sizeof arr` is 7.
- (b) `sizeof arr` is 8.
- (c) `sizeof ptr` is 8 on a typical 64-bit system.
- (d) `sizeof "SCS1301"` is 8.
- (e) `arr[0] = 'X';` is valid but `ptr[0] = 'X';` is undefined behaviour.

19. What is the output of the following code?

```
#include <stdio.h>
int main() {
    int a[6] = {5,10,15,20,25,30};
    int *p = a + 1, *q = a + 4;
    printf("%ld %d %d\n", (long)(q - p), *(p + 2), *q - *p);
    return 0;
}
```

- (a) 3 20 15
- (b) 3 15 15
- (c) 12 20 15
- (d) 3 20 10
- (e) Compilation error

20. Which of the following is/are correct regarding the program given below?

```
#include <stdio.h>
int main() {
    int m[3][4] = {{1,2,3,4},{5,6,7,8},{9,10,11,12}};
    int *flat = &m[0][0];
    printf("%d %d %zu\n", m[2][1], flat[9], sizeof m[0]);
    return 0;
}
```

- (a) m[2][1] is 10.
- (b) flat[9] is 10.
- (c) sizeof m[0] is 16.
- (d) sizeof m[0] is 48.
- (e) C stores the array in row-major order.

21. What is the value returned by g(13) for the function given below?

```
int g(int n){ if(n == 0) return 0; return g(n / 2) + n % 2; }
```

- (a) 1
- (b) 2
- (c) 3
- (d) 4
- (e) 13

22. Which of the following statement(s) is/are accurate about static variables in C?

- (a) Static variables are allocated from the heap.
- (b) Static variables are initialised to zero if not explicitly initialised.
- (c) Static variables are placed in the data segment rather than the stack.
- (d) A static local variable retains its value between calls to the function.
- (e) A static function is visible to other translation units.

23. Which of the following statement(s) is/are accurate regarding storage classes in C?

- (a) auto variables are stored in the stack and are valid within the local scope.
- (b) extern declares a name defined elsewhere and allocates no memory.
- (c) The address of a register variable can always be taken with &.
- (d) register is a request that the compiler may ignore.
- (e) static at file scope gives a name internal linkage.

24. Which of the following is/are correct regarding the memory layout of a C program?

- (a) The heap grows towards higher addresses and the stack towards lower addresses.
- (b) String literals are typically placed in a read-only section.
- (c) Uninitialised global variables are placed in the bss segment.
- (d) The bss segment occupies space in the executable file proportional to its size.
- (e) Local variables of a function are destroyed when the function returns.

25. Which of the following is/are correct regarding the program given below?

```
#include <stdlib.h>
int *make(void) {
    int v = 42;
    return &v;      // Line X
}
```

- (a) Line X returns the address of a local variable.
- (b) The returned pointer is dangling once make returns.
- (c) The program is correct because v is initialised.
- (d) Declaring v as static would make the returned pointer valid.
- (e) Allocating v with malloc would make the returned pointer valid.

26. Which of the following is/are correct regarding the program snippet given below?

```
struct P { char c; short s; int i; };
struct Q { int i; char c[3]; };
```

- (a) sizeof(struct P) is 8 on a typical 64-bit system.
- (b) sizeof(struct P) is 7.
- (c) sizeof(struct Q) is 8 on a typical 64-bit system.
- (d) Padding may be inserted after c in struct P.
- (e) Padding may be added at the end of struct Q.

27. Which of the following code snippets form a part of defining a pointer to a structure (structure A with members x and y), assigning the address of a structure to the pointer, and accessing a member through the pointer?

- (a) struct A *ptr;
- (b) struct A ptr*;
- (c) ptr = &temp;
- (d) ptr -> x;
- (e) (*ptr).x;

28. Which of the following is/are correct regarding file handling in C?
- (a) `fgetc` returns an `int` so that all byte values and EOF are distinguishable.
 - (b) `fopen` returns `NULL` if the file cannot be opened.
 - (c) `rewind(fp)` is equivalent to `fseek(fp, 0, SEEK_SET)` except that it also clears the error indicator.
 - (d) `while(!feof(fp))` is the recommended way to loop over a file.
 - (e) Writing at a position obtained with `fseek` inserts bytes and shifts the remainder of the file.
29. Which of the following sorting algorithm(s) is/are **out-place**, that is, require(s) $O(n)$ auxiliary memory proportional to the input size?
- (a) Bubble Sort
 - (b) Insertion Sort
 - (c) Merge Sort
 - (d) Quick Sort
 - (e) Selection Sort
30. Consider the array 29, 10, 14, 37, 13. Which of the following is/are correct?
- (a) After the second pass of insertion sort the array is 10, 14, 29, 37, 13.
 - (b) After the second pass of bubble sort the array is 10, 14, 13, 29, 37.
 - (c) After the second pass of selection sort the array is 10, 13, 14, 29, 37.
 - (d) Insertion sort is adaptive and runs in $O(n)$ on already sorted input.
 - (e) Selection sort performs fewer swaps than bubble sort on this array.
31. How many character comparisons would the Boyer–Moore algorithm perform to obtain the first match given the text COMPILER AND LINKER and the pattern LINKER?
- (a) 6
 - (b) 7
 - (c) 8
 - (d) 9
 - (e) 10
32. Which of the following statement(s) is/are correct regarding the naive string matching algorithm and the Boyer–Moore algorithm?
- (a) The naive algorithm compares the pattern left to right at every alignment.
 - (b) The naive algorithm has a worst case of $O(nm)$ for text length n and pattern length m .
 - (c) Boyer–Moore scans the alignment window from right to left.
 - (d) Boyer–Moore never performs more comparisons than the naive algorithm on any input.
 - (e) Boyer–Moore preprocesses the pattern before scanning the text.

33. Which of the following statement(s) is/are correct regarding stacks and their applications?

- (a) A stack follows the LIFO principle.
- (b) Balanced parenthesis checking can be solved using a stack.
- (c) Converting infix notation to postfix notation uses a stack.
- (d) peek removes the top element and returns it.
- (e) Recursive function calls are managed using a stack.

34. Which of the following is/are correct regarding doubly and circular linked lists?

- (a) A doubly linked list node stores two pointers, increasing memory per node.
- (b) Deleting a known node from a doubly linked list is $O(1)$.
- (c) In a circular linked list the last node points back to the head.
- (d) Traversing a circular list with `while(p != NULL)` terminates correctly.
- (e) A doubly linked list allows traversal in both directions.

35. Which of the following statement(s) is/are (an) equivalent to decreasing an integer variable x value by one?

- (a) `x -= 1`
- (b) `x -= 1`
- (c) `x--`
- (d) `--x`
- (e) `x = x - 1`

PART II

Answer the question given below. The question carries thirty (30) marks.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  struct Node {
5      int data;
6      struct Node *next;
7  };
8
9  struct Node *A(int value){
10     struct Node *n = malloc(sizeof(struct Node));
11     n->data = value;
12     n->next = NULL;
13     return n;
14 }
15
16 void B(struct Node **head, int value){
17     struct Node *n = A(value);
18     if(*head == NULL){ *head = n; return; }
19     struct Node *p = *head;
20     while(p->next != NULL) p = p->next;
21     p->next = n;
22 }
23
24 struct Node *C(struct Node *head){
25     struct Node *prev = NULL, *cur = head, *nxt;
26     while(cur != NULL){
27         nxt = cur->next;
28         cur->next = prev;
29         prev = cur;
30         cur = nxt;
31     }
32     return prev;
33 }
34
35 int main(int argc, char *argv[]){
36     struct Node *head = NULL;
37     for(int i = 1; i < argc; i++){
38         B(&head, atoi(argv[i]));
39     }
40     head = C(head);
41
42     struct Node *p = head;
43     while(p != NULL){ printf("%d ", p->data); p = p->next; }
44     printf("\n");
45     return 0;
46 }
```

Consider the program given above in lines 1 to 45, which processes a sequence of integers provided as command-line arguments, and answer the questions (a) to (d) given below.

(a) Explain the core functionality of the functions A (3 marks), B (4 marks) and C (5 marks). Diagrams are required where necessary, and relevant line numbers are to be referred to in your explanation. **[12 Marks]**

(b) Explain the issues (including errors) expected when executing the code above and propose solutions to the problems identified. **[05 Marks]**

(c) What would be the output if the program was executed with the arguments 12 7 25 3? Show the state of the list after the call in line 39. **[05 Marks]**

(d) Write a function named `struct Node *D(struct Node *head, int value)` using the C programming language that deletes the *first* node containing value from the list, frees it, and returns the head of the resulting list. Your function must handle the cases where the list is empty, where the node to be deleted is the head, and where the value is not present. **[08 Marks]**

———— End of Paper ————