

BACHELOR OF SCIENCE IN COMPUTER SCIENCE

First Year Examination – Semester I

SCS 1301 – Data Structures and Program Design in C

Sample Paper A

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 11 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 the C preprocessor?
 - (a) Comments are removed before macro expansion is complete.
 - (b) It performs textual substitution and does not parse C grammar.
 - (c) Macro definitions are terminated by a semicolon.
 - (d) The output of the preprocessor is still C source code.
 - (e) `#include` directives are replaced by the contents of the named file.

2. Which of the following statement(s) is/are true regarding the GCC compilation process?
 - (a) The option `-c` stops the process after the assembler and produces a `.o` file.
 - (b) The option `-E` stops the process after preprocessing.
 - (c) The option `-S` produces an output file with the extension `.s`.
 - (d) undefined reference to `'sqrt'` is produced by the compiler, not the linker.
 - (e) Without `-o`, the linker produces an executable named `a.out`.

3. Which of the following is/are guaranteed by the C standard?
 - (a) `sizeof(char)` is 1.
 - (b) `sizeof(int)` is 4.
 - (c) `sizeof(long) >= sizeof(int)`.
 - (d) A float occupies fewer bytes than a double.
 - (e) Signed integer overflow wraps around to the minimum value.

4. Which of the following contain the correct operator precedence from the highest to the lowest?
Do not consider the comma as an operator for this question.
 - (a) `[]`, `*`, `+`
 - (b) `<<`, `%`, `++(prefix)`
 - (c) `&&`, `||`, `=`
 - (d) `^`, `&`, `==`
 - (e) `sizeof`, `?:`, `+=`

Consider the following header file `"defs.h"` and the associated source file `"test.c"` to answer the Questions 5 to 10.

----- 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 i, j, k;
5
6     i = j = 0;
7     for(i = 0; i < 5; i++) j += i;
8     PRINT2(d,i,j);
9
10    i = 0; k = 0;
11    while(i++ < 5) k += i;
12    PRINT2(d,i,k);
13
14    i = 0; j = 0;
15    do { j += ++i; } while(i < 3);
16    PRINT2(d,i,j);
17
18    i = 5; j = 0;
19    for(; i; i >=> 1) j++;
20    PRINT2(d,i,j);
21
22    for(i = 0; i < 3; i++)
23        for(j = 0; j < 3; j++)
24            if(j == 1) break;
25    PRINT2(d,i,j);
26 }
```

5. Which of the following is/are (a) correct expansion(s) of the macro call PRINT2(d,i,j)?

- (a) PR(d, i), PR(d, j), NL
- (b) PR(d, i), PRINT1(d, j)
- (c) printf("i =%d\t", (i)), printf("j =%d\t", (j)), putchar('\n')
- (d) printf("#i " =%" #d "\t", (i)), NL
- (e) printf("i =%d\t", (i)), putchar('\n'), printf("j =%d\t", (j))

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

- (a) i = 4
- (b) i = 5
- (c) j = 10
- (d) j = 15
- (e) j = 20

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

- (a) i = 5
- (b) i = 6
- (c) k = 10
- (d) k = 15
- (e) k = 21

8. Which of the following is/are (a) value(s) printed by the statement in line 16?

- (a) i = 2
- (b) i = 3
- (c) j = 3
- (d) j = 6
- (e) j = 10

9. Which of the following is/are (a) value(s) printed by the statements in lines 20 and 26?

- (a) Line 20 prints i = 0
- (b) Line 20 prints j = 3
- (c) Line 20 prints j = 5
- (d) Line 26 prints i = 3
- (e) Line 26 prints j = 1

Consider the header file "defs.h" given above and the associated source file given below and answer the Questions 11 to 15.

```
1 #include "defs.h"
2
3 int a[] = {2, 4, 6, 8, 10};
4 int *p[] = {a, a+1, a+2, a+3, a+4};
5 int **pp = p;
6
7 int main(){
8     PRINT3(d,*a,**p,**pp);
9
10    pp += 2;    PRINT3(d,pp-p,**pp-a,**pp);
11    ++*pp;      PRINT3(d,pp-p,**pp-a,**pp);
12    ++**pp;     PRINT3(d,pp-p,**pp-a,**pp);
13    NL;
14
15    pp = p;
16    *++pp;      PRINT3(d,pp-p,**pp-a,**pp);
17    *pp += 2;   PRINT3(d,pp-p,**pp-a,**pp);
18    **pp += 5;  PRINT3(d,pp-p,**pp-a,**pp);
19 }
```

10. Which of the following is/are correct regarding the declarations in lines 3 to 5?

- (a) p is an array of five pointers to int.
- (b) p is a pointer to an array of five int.
- (c) pp is a pointer to a pointer to int.
- (d) sizeof(a) and sizeof(p) are equal on a 64-bit system.
- (e) **pp and a[0] refer to the same object before main runs.

11. What would be the output of line 8?

- (a) `*a = 2    **p = 2    **pp = 2`
- (b) `*a = 2    **p = 4    **pp = 6`
- (c) `*a = 0    **p = 0    **pp = 0`
- (d) `*a = 10   **p = 10   **pp = 10`
- (e) Garbage values, because pp has not been advanced.

12. Which of the following is/are (a) value(s) printed by the statement in line 11?

- (a) `pp-p = 2`
- (b) `pp-p = 3`
- (c) `*pp-a = 2`
- (d) `*pp-a = 3`
- (e) `**pp = 8`

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

- (a) `pp-p = 2`
- (b) `*pp-a = 3`
- (c) `*pp-a = 4`
- (d) `**pp = 8`
- (e) `**pp = 9`

14. Which of the following is/are correct regarding the statements in lines 17 and 18?

- (a) Line 17 modifies an element of the array a.
- (b) Line 17 modifies an element of the array p.
- (c) Line 17 prints `*pp-a = 3`.
- (d) Line 18 modifies an element of the array a.
- (e) Line 18 prints `**pp = 14`.

15. Which of the following are accurate asymptotic complexities for the worst case of inserting an element at the head of a singly linked list of n nodes?

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

16. You are given a sorted array of n elements and you perform binary search. Which are correct worst-case asymptotic complexities?

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

17. Consider the loop `for(int i = 1; i < n; i = i * 3) { /* O(1) body */ }`. Which are accurate asymptotic complexities?

- (a) $O(\log n)$
- (b) $O(\sqrt{n})$
- (c) $O(n)$
- (d) $\Theta(\log n)$
- (e) $\Theta(n)$

18. Which of the following statement(s) is/are true regarding `void *` in C?

- (a) A `void *` can be assigned the address of any object type without an explicit cast.
- (b) `malloc` returns a `void *`.
- (c) Pointer arithmetic on a `void *` is defined by the C standard.
- (d) A `void *` can be dereferenced directly without a cast.
- (e) A `void *` discards type information, so the programmer must track the type.

19. What is `X` in `char *(*X[5])(int, char *)`?

- (a) `X` is an array of 5 pointers to functions passing an `int` and a pointer to `char`, returning a pointer to `char`.
- (b) `X` is a pointer to an array of 5 functions passing an `int` and a pointer to `char`, returning a pointer to `char`.
- (c) `X` is a function passing an `int` and a pointer to `char`, returning an array of 5 pointers to `char`.
- (d) `X` is an array of 5 functions returning pointers to `char`.
- (e) `X` is a pointer to a function passing an `int` and a pointer to `char`, returning an array of 5 pointers to `char`.

20. Which of the following statements are true about `ptr` if it was declared as `int * const ptr = &x;`?

- (a) The pointer must be initialised at the point of declaration.
- (b) The pointer can later be made to point at a different object.
- (c) The value at the memory location can be changed through `ptr`.
- (d) `*ptr = 10;` is a compilation error.
- (e) `ptr = &y;` is a compilation error.

21. What is the output of the following code?

```
#include <stdio.h>
int main() {
    char data[] = {'A','B','C','D','E'};
    void *vp = data;
    vp = (char *)vp + 3;
    printf("%c\n", *(char *)vp);
    return 0;
}
```

- (a) A
- (b) C
- (c) D
- (d) Compilation error
- (e) Undefined behaviour

22. What is the output of the following code?

```
#include <stdio.h>
void counter(void){ static int c; c++; printf("%d ", c); }
int main() {
    counter(); counter(); counter();
    return 0;
}
```

- (a) 1 1 1
- (b) 1 2 3
- (c) 0 1 2
- (d) Garbage values, because c is not initialised.
- (e) Compilation error

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

```
#include <stdio.h>
void f(int arr[10]){ printf("%zu\n", sizeof arr); }
int main() {
    int arr[10];
    printf("%zu\n", sizeof arr);
    f(arr);
    return 0;
}
```

- (a) The program prints 40 then 8 on a typical 64-bit system.
- (b) The program prints 40 twice.
- (c) Inside f, arr is a pointer, not an array.
- (d) The 10 in the parameter declaration is ignored by the compiler.
- (e) A function can determine the length of an array passed to it using sizeof.

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

```
#include <stdio.h>
#include <stdlib.h>
int main() {
    int *p = malloc(4 * sizeof(int));
    p = realloc(p, 8 * sizeof(int));    // Line X
    free(p);
    return 0;
}
```

- (a) If realloc fails at Line X, the original block is leaked.
- (b) realloc may return a different address from the one passed in.
- (c) The return value of malloc should be checked before Line X.
- (d) Using a temporary pointer for the result of realloc avoids the leak.
- (e) free(p) is unnecessary because realloc already freed the old block.

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

```
#include <stdlib.h>
int main() {
    int *q = malloc(sizeof(int));
    free(q);
    free(q);    // Line Y
    return 0;
}
```

- (a) Line Y is a double free and is undefined behaviour.
- (b) Setting q = NULL; after the first free makes Line Y harmless.
- (c) free(NULL) is defined to do nothing.
- (d) After the first free, q is a dangling pointer.
- (e) The program leaks memory.

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

```
struct S1 { char a; double b; char c; };
struct S2 { double b; char a; char c; };
```

- (a) sizeof(struct S1) is 24 and sizeof(struct S2) is 16 on a typical 64-bit system.
- (b) Both structures have the same size because they have the same members.
- (c) Padding may be inserted between members to satisfy alignment.
- (d) Padding may be added at the end of a structure.
- (e) Declaring members in decreasing order of size tends to reduce padding.

27. Which of the following is/are correct regarding unions?

```
union V { char s[10]; int i; double d; };
```

- (a) All members may hold meaningful values at the same time.
- (b) `sizeof(union V)` is at least 10.
- (c) `sizeof(union V)` is 16 on a typical 64-bit system.
- (d) Writing to `i` and then reading `d` reinterprets the same bytes.
- (e) The primary purpose of a union is to save memory.

28. Which of the following is/are correct regarding the content of `log.txt` after the program snippet below is executed twice in succession?

```
FILE *fp = fopen("log.txt", "w");  
fprintf(fp, "AB");  
fclose(fp);  
fp = fopen("log.txt", "a");  
fprintf(fp, "CD");  
fclose(fp);
```

- (a) After the first execution the file contains ABCD.
- (b) After the second execution the file contains ABCDABCD.
- (c) After the second execution the file contains ABCD.
- (d) Opening in mode "w" truncates the file at the moment of opening.
- (e) Opening in mode "a" creates the file if it does not exist.

29. Which of the following sorting algorithm(s) is/are stable?

- (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 one complete pass of bubble sort the array is 10, 14, 29, 13, 37.
- (b) After one complete pass of bubble sort the array is 10, 29, 14, 37, 13.
- (c) After one complete pass of selection sort the array is 10, 29, 14, 37, 13.
- (d) Bubble sort performs five (05) swaps in total on this array.
- (e) Selection sort performs three (03) swaps in total on this array.

31. How many character comparisons would the Boyer–Moore algorithm perform to obtain the first match given the text THE QUICK BROWN FOX and the pattern BROWN?
- (a) 5
 - (b) 6
 - (c) 7
 - (d) 8
 - (e) 9
32. Which of the following statement(s) is/are true regarding the Boyer–Moore algorithm?
- (a) Characters within the current alignment window are compared right to left.
 - (b) The last occurrence table stores the rightmost index of each pattern character.
 - (c) For a character not present in the pattern the last occurrence value is -1 .
 - (d) The algorithm may shift the pattern backwards to avoid missing a match.
 - (e) A larger alphabet generally allows larger shifts.
33. Which of the following statement(s) is/are correct regarding queues?
- (a) A queue follows the FIFO principle.
 - (b) A linear array queue can report itself full while cells at the front are unused.
 - (c) In a circular queue the index advances as $(\text{index} + 1) \% \text{CAPACITY}$.
 - (d) Both enqueue and dequeue are $O(1)$ in a circular array queue.
 - (e) A queue requires only a single pointer to operate.
34. Which of the following is/are correct regarding a singly linked list with head and tail pointers?
- (a) AddToHead is $O(1)$.
 - (b) AddToTail is $O(1)$.
 - (c) DeleteFromTail is $O(1)$.
 - (d) Deleting the only node in the list must set both head and tail to NULL.
 - (e) Accessing the k^{th} element is $O(1)$.
35. Which of the following statement(s) is/are (an) equivalent to multiplying an integer variable x by two, assuming x is a non-negative int?
- (a) $x *= 2$
 - (b) $x = x << 1$
 - (c) $x <=& 1$
 - (d) $x = x + x$
 - (e) $x =* 2$

PART II

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

(a) Explain how you would implement a **circular queue** using an array and a **queue** using a singly linked list. Your answer must give the structure definitions, the enqueue and dequeue operations, and the tests for the empty and full conditions in each case. Diagrams are required where necessary, and all special cases must be handled. **[18 Marks]**

(b) Compare the two implementations in terms of memory usage, the time complexity of each operation, and the flexibility of the base structure. Your answer must state the specific problem that circularity solves in the array implementation, and explain why that problem does not arise in the linked list implementation. **[12 Marks]**

———— End of Paper ————