

②

ISLAMIC UNIVERSITY OF TECHNOLOGY (IUT)
ORGANISATION OF ISLAMIC COOPERATION (OIC)
Department of Computer Science and Engineering (CSE)

MID SEMESTER EXAMINATION
 DURATION: 1 HOUR 30 MINUTES

SUMMER SEMESTER, 2022-2023
 FULL MARKS: 75

CSE 4649: Systems Programming

Programmable calculators are not allowed. Do not write anything on the question paper.
 Answer all 3 (three) questions. Figures in the right margin indicate full marks of questions whereas corresponding CO and PO are written within parentheses.
 [For all the questions, assume **64-bit** system unless otherwise mentioned]

1. a) Consider the following C program in Code Snippet 1.

```

1 int main() {
2     unsigned int ui = -2;
3     unsigned short usi = 69;
4     char c = 0x80;
5     if (~!! > ui)
6         printf("Hello\n");
7     else
8         printf("World\n");
9     if (c > usi)
10        printf("Hasta\n");
11    else
12        printf("Vista\n");
13 }
  
```

Code Snippet 1: A C program for Question 1.a

Mention the output of the program given in Code Snippet 1. Explain the output with proper reasoning by mentioning the type conversion rules being applied in each conditional expression. Assume the code will be run in a 32-bit machine.

- b) Considering the C program in Code Snippet 2, identify and explain the bugs with proper reasoning.

```

1 int main() {
2     char *my_str = NULL;
3     printf("my_str is %s\n", *my_str);
4
5     char *hello = "Hi!";
6     hello[3] = '!';
7     printf("%s\n", hello);
8
9     int *arr = malloc(sizeof(int) * 69);
10    for (int i = 1; i < 69; i++) {
11        arr[i - 1] = i;
12    }
13    printf("%d\n", arr[68]);
14 }
  
```

Code Snippet 2: A buggy C program for Question 1.b

- c) Explain the role of Instruction Set Architecture (ISA) in maintaining Binary Compatibility with appropriate examples.

12
(CO1)
(PO1)

6
(CO2)
(PO2)

7
(CO1)
(PO1)

2. Consider the C program in Code Snippet 3 for answering Question 2.

```
1 void print_arr_2d(int nr, int nc, int m[][nc]) {
2     for (int row = 0; row < nr; row++) {
3         for (int col = 0; col < nc; col++)
4             printf("0x%x\t", m[row][col]);
5         printf("\n");
6     }
7 }
8 int main() {
9     int num_rows = 2, num_cols = 3;
10    int arr_2d[num_rows][num_cols];
11    for (int row = 0; row < num_rows; row++)
12        for (int col = 0; col < num_cols; col++)
13            arr_2d[row][col] = 0x69 + row;
14    print_arr_2d(num_rows, num_cols, arr_2d);
15 }
```

Code Snippet 3: A C program to print 2D array for Question 2

a) Mention the output of the program given in Code Snippet 3.

3
(CO1)
(PO1)

b) Suppose the `modify_arr_2d()` function in Code Snippet 4 has been added to the program in Code Snippet 3.

4 + 8
(CO1)
(PO1)

```
1 void modify_arr_2d(int nr, int nc, int m[][nc]) {
2     short *p = &m[0][1];
3     p++;
4     *p = 0x1569;
5     *(p + 2) = 0x69;
6     p[4] = 0x6900;
7 }
```

Code Snippet 4: `modify_arr_2d()` function for Question 2b

The `main()` function is modified by adding the lines given in line number 14 of Code Snippet 3.

```
1 modify_arr_2d(num_rows, num_cols, arr_2d);
2 print_arr_2d(num_rows, num_cols, arr_2d);
```

Code Snippet 5: `main()` after `modify_arr_2d()` function for Question 2b

Draw the memory layout of `arr_2d` without calling and after calling `modify_arr_2d()` function. Consider the base of the array starts at address `0x62fd70`. The layout should include the address of each individual byte locations and the values in those addresses in hexadecimal. The first two byte locations, before call to `modify_arr_2d()`, are done for you.

```
// Before call to modify_arr_2d()
| Address | Value |
+-----+-----+
+ 0x62fd70 + 0x00 +
+ 0x62fd71 + 0x00 +
```

c) What will be the output of `print_arr_2d()` at line no. 2 of Code Snippet 5 after call to `modify_arr_2d()`? 5
(CO1)
(PO1)

d) State whether the following statements are correct or incorrect. 5
(CO1)
(PO1)

- i. For the array declaration `int arr[10][8]`, `sizeof(*arr)` will return 8.
- ii. For the array declaration `int arr[10]`, `arr++` will make `arr` point to the second element of the array.
- iii. For the variable declaration `unsigned int x = 0xfffffbb`, `printf("%d", x);` will output -69.
- iv. Intel uses bi-endianness in all of its processors.
- v. `pause()` suspends the execution of the calling process until a signal arrives.

3. a) Draw the process graph of the C program given in Code Snippet 6. 8
(CO1)
(PO1)

```
1 int main() {
2     int i;
3     int pid = fork();
4     for (i = 0; i < 2; i++) {
5         if (pid == 0) {
6             fork();
7         }
8     }
9     printf("hello\n");
10    exit(0);
11 }
```

Code Snippet 6: A C program for Question 3.a

b) Write a C program using `execve()` that would change the current working directory to the parent directory and print the contents of that parent directory. The required commands (`cd` and `ls`) must be executed in a child process. After the child process is finished, the execution should resume in the original parent process. 8
(CO3)
(PO2)

c) Consider the definition of a struct named `student` in Code Snippet 7. 3
(CO1)
(PO1)

```
1 typedef struct student {
2     long id;
3     char name[20];
4     int is_resident;
5     short *next;
6 } student_t;
7 student_t arr[10];
```

Code Snippet 7: A struct for Question 3.c

Assume the beginning of the array `arr` is at byte offset of zero. What is the byte offset from the beginning of the array for the following things:

- i. the third student struct in the array
- ii. the third character in the `name` field, in the second student struct in the array
- iii. the `next` field in the fourth student struct in the array

d) Write two possible sequence of outputs for the following C program in Code Snippet 8.

6
(CO2)
(PO2)

```
1 int main() {
2     int x = 69;
3     pid_t pid_1, pid_2;
4     pid_1 = fork();
5     if (pid_1 == 0) {
6         x++;
7         pid_2 = fork();
8         if (pid_2 == 0) {
9             printf("p1: %d\n", x++);
10            fflush(stdout);
11            exit(0);
12        }
13    } else {
14        printf("p2: %d\n", x--);
15        fflush(stdout);
16    }
17    printf("p3: %d\n", x);
18    fflush(stdout);
19 }
```

Code Snippet 8: A C program for Question 3.d