Princess Sumaya University for Technology
جامعـــة الأميـــرة سميّة للتكنولوجيا

King Hussein School for Computing Sciences
Department of Computer Science
11103 - **Structured Programming** - Spring 2023

Final Exam

Full Name: **Student ID:**

Question	Marks	Score
1	5	
2	12	
3	3	
4	6	
5	7	
6	7	
Total	40	

Circle your section:

○ Dr. Ammar Alrashdan	(section 1)
○ Dr. Osama Alhaj Hasan	(section 2)
○ Dr. Rawan Ghnemat	(section 3)
○ Dr. Ammar Alrashdan	(section 4)
○ Dr. Rawan Ghnemat	(section 5)
○ Dr. Mohammad Al Nabhan	(section 6)
○ Dr. Mohammad Al Nabhan	(section 7)
○ Dr. Manaf Gharaibeh	(section 8)
○ Dr. Mohammad Abu Snober	(section 9)
○ Dr. Mohammad Abu Snober	(section 10)
○ Mr. Yousef Yaseen	(section 11)
○ Mr. Alaa Altarazi	(section 12)
○ Mr. Alaa Altarazi	(section 13)
○ Mr. Alaa Altarazi	(section 14)

12/6/2023

Question 1. Basics (5 marks)

```
 1. double x = 5;
 2.
 3. // ---- THIS LINE ---- //
 4.
 5. int main() {
 6.     x++;
 7.     double x = 2.5;
 8.     f(x);
 9.     printf("%f\n", x);
10.     return 0;
11. }
```

A. [3 marks] For each of the following function definitions, show the output of the above program if the function definition is placed at line #3. If the code causes a compile-time or a run-time error, write "**ERROR**".

FUNCTION DEFINITION TO REPLACE LINE # 3	PROGRAM OUTPUT
1. **void** f(**double** x) { x = 3; }	2.500000 (2.5 is OK)
2. **void** f(**double*** x) { *x = 3; }	ERROR
3. **void** f(**double** x) { **int** x = 3; }	ERROR
4. **void** f(**double** y) { x = 3; }	2.500000 (2.5 is OK)
5. **void** f(**double** y) { printf("%f ", x); }	6.000000 2.500000 (6.0 and 2.5 is OK)
6. **void** f(**int** y) { printf("%d ", y); }	2 2.500000 (2 and 2.5 is OK)

B. [2 marks] Assuming that x is defined as **int** x = 1; what is the output of each of the following pieces of code? If the code causes a compile-time or a run-time error, write "**ERROR**".

1.
```
printf("%d", x + 1 / 2);
```
1

2.
```
if   (x  > 2) printf("YES");
else (x <= 2) printf("NO");
```
ERROR

3.
```
if (x = 2) printf("2");
else       printf("1");
```
2

4.
```
switch (rand() % x) {
 case 0:  printf("0 ");
          break;
 default: printf("NO"); }
```
0

Question 2. Mini Code Writing (12 marks)

Implement each of the following functions.

(A)

```
// Returns 1 if at least one argument is positive and at least one argument is
// negative, and 0 otherwise.
int diff_sign(int a, int b, int c) {

    return !(a > 0 && b > 0 && c > 0) && !(a < 0 && b < 0 && c < 0);

}
```

(B)

```
// Prints the pattern: 1 100 2 99 3 98 4 97 5 96 ... 99 2 100 1
void print_pattern() {

    for (int i = 1, j = 100; i <= 100; i++, j--) {
        printf("%d %d ", i, j);
    }

}
```

(C)

```
// Prints the pattern: 1-  2--  3---  4----  5-----  ...         (assuming n > 0)
int n_dashes(int n) {         // the last term printed is n followed by n dashes
    for (int i = 1; i <= n; i++) {
        printf("%d", i);
        for (int j = 0; j < i; j++)
            printf("-");
        printf(" ");
    }
}
```

(D)

```
// Returns 1 if every row in a[][] sums to 50. Returns 0 otherwise.
int sum_50(int a[10][10]) {

    for (int i = 0; i < 10; i++) {
        int sum = 0;
        for (int j = 0; j < 10; j++)
            sum += a[i][j];
        if (sum != 50)
            return 0;
    }
    return 1;
}
```

Question 3. Code Comprehension (3 Marks)

```c
void fun(int a[], int N) {
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < N; j++)
            printf("A B ");

        int sum = 0;
        for (int j = 0; j < 100; j++) {
            sum += a[j];
            printf("B ");
        }

        a[i] = sum;
        printf("%d\n", N / sum);
    }
}
```

Answer the questions below about the above function.

A. `[1 mark]` Assume that `N=100`, and that `a[]` is of size `100`.

- How many times will "**A**" be printed out? 10000
- How many times will "**B**" be printed out? 20000

B. `[1 mark]` Mention two cases that might cause a run time error.

Case 1: If sum = 0 (printf("%d", N / sum) will crash)

Case 2: If the size of the array is less than 100 (sum += a[j] might crash)

C. `[1 mark]` Assume that `N=100`, and `a[]` is of size `100` and every cell in `a[]` contains the value `1`. What will be stored at `a[0]` after the function finishes execution?

100

Question 4. Recursion (6 marks)

1. [4 marks] Implement function `int remainder(int n, int m)` which returns `n % m`.

Notes.

- The function must be recursive. You are not allowed to use loops.
- You are not allowed to use the `%` or `/` operators.
- You can assume that `n` and `m` are greater than `0`.

2. Answer the questions on the right about the following function.

```
void fun(int n) {
   if (n <= 0)
       return;
   fun(n-1);
   fun(n-2);

   printf("%d ", n);
}
```

1. [0.5 mark] What is the output of calling `fun(1)` ?

```
1
```

2. [0.5 mark] What is the output of calling `fun(2)` ?

```
1 2
```

3. [1 mark] What is the output of calling `fun(4)` ?

```
1 2 1 3 1 2 4
```

Question 5. Strings and Pointers (7 marks)

Implement function `void remove_duplicates(char* str, char* result)`, which copies `str` to `result` after removing duplicate characters that are next to each other.

Examples.

str	**result**
`aaabbbaaa`	`aba`
`ababab`	`ababab`
`helloooo, there!!!`	`helo, there!`

Notes.

- You can assume that no string will be longer than `100` characters.
- You are not allowed to use the array `[]` notation. You must use pointer arithmetic only.
- You are not allowed to use the `string.h` library.

Provide your answer on the following page

```
void remove_duplicates(char* str, char* result) {
    char* ptr1 = str;
    char* ptr2 = result;

    while (*ptr1 != '\0') {
        if (*ptr1 != *(ptr1 + 1)) {
            *ptr2 = *ptr1;
            ptr2++;
        }
        ptr1++;
    }

    *ptr2 = '\0';
}
```

Question 6. 2D Arrays (7 marks)

Implement function `void print(int a[N][N])`, which prints the received 2D square matrix as follows:

```
1st row elements
1st column elements

2nd row elements
2nd column elements

3rd row elements
3rd column elements

etc.
```

Examples.

```
ARRAY      OUTPUT           ARRAY  OUTPUT         ARRAY      OUTPUT
1 2 3 4    1 2 3 4          1 2    1 2            1          1
5 6 7 8    1 5 9 3          3 4    1 3                       1
9 0 1 2
3 4 5 6    5 6 7 8                 3 4
           2 6 0 4                 2 4

           9 0 1 2
           3 7 1 5

           3 4 5 6
           4 8 2 6
```

You can assume that N is globally defined and accessible.

Provide your answer for **Q6** **only** in the **Answers Booklet**

```c
void print(int a[N][N]) {

    for (int i = 0; i < N; i++) {

        for (int j = 0; j < N; j++)
            printf("%d ", a[i][j]);

        printf("\n");

        for (int j = 0; j < N; j++)
            printf("%d ", a[j][i]);

        printf("\n\n");
    }
}
```