



Princess Sumaya جامعة
University الأميرة سميرة
for Technology للتكنولوجيا

The King Hussein School for Computing Sciences
Department of Computer Science
Structured Programming - Spring 2022

Final Exam

Full Name:

Student ID:

Circle your section:

- Dr. Mu'awya Al-Dala'ien (section 1)
- Dr. Rawan Ghnemat (section 2)
- Dr. Abdullah Aref (section 3)
- Dr. Mu'awya Al-Dala'ien (section 4)
- Dr. Rawan Ghnemat (section 5)
- Dr. Sawsan Alshatnawi (section 6)
- Dr. Mohammad Al Nabhan (section 7)
- Dr. Sawsan Alshatnawi (section 8)
- Dr. Mohammad Abu Snober (section 9)
- Dr. Mohammad Abu Snober (section 10)
- Dr. Khaled Mansour (section 11)
- Dr. Abedalrhman Alkhateeb (section 13)
- Dr. Khaled Mansour (section 14)
- Dr. Rafat Hammad (section 15)

Question	Points	Score	
1	16		
2	30		
3	40	PART 1 (14)	
		PART 2 (16)	
		PART 3 (10)	
4	14		
Total	100		

Saturday 18/6/2022

Question 1. Mini-Tracing (16 points)

expected time = 5 - 15 minutes

Fill the **Output** column in the table below with the output of the code provided in the **Code** column. If the code does not compile, write "**compilation error**" instead of the output.

Assume that x, y and a are **global variables** defined as: **int** x = 1, y = 2, a[3] = {5, 1, 9};

Code	Output
<pre>1. for (int i = 0; i < 3; i++) printf("%d", i); printf("%d", i+1);</pre>	
<pre>2. switch (x) { case 1: printf("A"); case 2: printf("B"); }</pre>	
<pre>3. for (int i = 0; i < 3; i++) a[i++]; for (int j = 0; j < 3; j++) printf("%d ", a[j]);</pre>	
<pre>4. printf("%d ", *(a+2));</pre>	
<pre>5. printf("%d ", *(++a));</pre>	
<pre>6. int* ptr = &x; x++; printf("%d", *ptr);</pre>	
<pre>7. void f(int c) { int c = 5; printf("%d", c); } int main() { f(7); return 0; }</pre>	
<pre>8. void fun(int* p1) { (*p1)++; printf("%d", *p1); } int main() { fun(3); return 0; }</pre>	

Question 2. Mini-Coding (30 points)

expected time = 10 - 30 minutes

Implement each of the following functions.

<p>(A)</p> <pre>// Returns 1 if a, b and c are the same // or if they are all different. // Returns 0 otherwise. int zor(int a, int b, int c) { } }</pre>	<p>(B)</p> <pre>// Returns the number of digits in x int digits(int x) { } }</pre>
<p>(C)</p> <pre>// Returns 1 if all the integers in a[] // are odd. Returns 0 otherwise. int all_odd(int a[], int size) { } }</pre>	<p>(D)</p> <pre>// Fills (only) the first and last // columns of the 20x20 array with 0 void fill_zeros(int a[20][20]) { } }</pre>
<p>(E)</p> <pre>// Replaces all the characters of the given string with 'X' void hide(char* str) { } }</pre>	

Question 3. Code Writing (40 points)

expected time = 20 - 60 minutes

PART 1. ARRAYS AND LOOPS (14 points)

Implement a **void** function named **fill_random(...)** that receives a 1D array and its size as arguments and fills it with 100000 random integers in the range $[-100000, 100000]$, provided that **no number appears more than once** in the array.

PART 2. STRINGS (16 points)

Implement a program that reads two strings from the user and merges them into a new string, as the following examples show. The program should then print the resulting string.

Examples.

```
string 1 = "ccccc"  
string 2 = "ggggg"  
result   = "cgcgcgcgcg"
```

```
string 1 = "XYZ"  
string 2 = "cccccc"  
result   = "XcYcZcccc"
```

```
string 1 = "00000000"  
string 2 = "TBA"  
result   = "0T0B0A00000"
```

```
string 1 = ""  
string 2 = "ABC"  
result   = "ABC"
```

Notes.

- You can assume that no string entered by the user is longer than 100 characters.
- **Define your strings as arrays of characters.** However, you must use pointer arithmetic when processing the strings. You are not allowed to use array notation anywhere other than when defining the strings.
- You are not allowed to use the `string.h` library.

PART 3. RECURSION (10 points)

Implement a **recursive** function named **plot(...)** that receives as arguments an array and its size, and plots a histogram of the integers that are in the array (see the example provided below).

Notes.

- You are allowed to use loops *with recursion*. You are **not** allowed to use loops *without* recursion.
- You are allowed to add extra parameters to the function if you need, provided that you explain in a comment what the parameters are.

Example.

Array:	a[] = {1, 5, 0, 3, 7}
Output:	1: * 5: ***** 0: 3: *** 7: *****

Question 4. Code Reading (14 points)

expected time = 5 - 15 minutes

PART 1. What is the output of the function below in each of the following cases?

```
void mystery(int N) {
    int a, b, c = 0;

    for (int i = 0; i < N; i++) {
        scanf("%d", &a);
        if (a % 2 == 0) {
            b = a;
            c++;
        }
    }

    if (c > 0)
        printf("%d", b);
}
```

A. If $N = 2$ and the user input is 4 7

B. If $N = 3$ and the user input is 1 1 1

C. If $N = 100$ and the user input is 1 2 3 4 ... 100

What is the purpose of (الهدف من) this code?

Do not use > 20 words in your answer (write in the box).

PART 2. What is the content of the array $a[][]$ after executing **mystery** in each of the following cases?

```
// assume N is a global constant
// defined and initialized here

void mystery(int a[N][N]) {
    for (int i = 0; i < N; i++) {
        int temp = a[i][i];
        a[i][i] = a[i][N-i-1];
        a[i][N-i-1] = temp;
    }
}
```

A. If $N=1$ and $a = \{1\}$

B. If $N=2$ and $a = \{\{1, 2\}, \{3, 4\}\}$

C. If $N=100$ and every row in the array contains $\{1, 2, 3, 4, 5, \dots, 97, 98, 99, 100\}$

show only the last 3 elements of the last 3 rows.

What is the purpose of (الهدف من) this code?

Do not use > 20 words in your answer (write in the box).

This page is intentionally left blank