



The King Hussein School for Computing Sciences
Department of Computer Science
Structured Programming - Fall 2021

Final Exam

Full Name:

Solutions

Student ID:

Instructions.

- There are **10** pages, including this cover page and a blank page at the end for you to use as a scratch paper. Inform your instructor if you have less than **10** pages.
- Write clearly. You may not receive credit for your answer if it is not readable.

Question	Points	Score
1	12	
2	28	
3	18	
4	24	
5	20	
Total	100	

Circle your section:

- Dr. Mohammad Al Nabhan (section 1)
- Dr. Rawan Ghnemat (section 2)
- Dr. Rawan Ghnemat (section 3)
- Dr. Mohammad Abu Snober (section 4)
- Dr. Abdullah Aref (section 5)
- Dr. Sawsan Alshatnawi (section 6)
- Dr. Mohammad Al Nabhan (section 7)

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

```
int main() {
    int N;
    int result = -1;
    scanf("%d", &N);

    for (int i = 1; i <= N; i++) {
        if (i*i == N)
            result = i;
    }

    printf("%d", result);
    return 0;
}
```

A. If $N = 1$

B. If $N = 2$

C. If $N = 400$

D. If $N = 401$

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

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

It prints the square root of N if the square root is an integer and prints -1 otherwise

PART 2. What are the contents of array `a[]` after executing the code below in each of the following cases?

```
// Assume that an array named
// a[] of size N is defined
// and initialized here

int i = 0, j = N-1;
while (i < j) {
    while (i < j && a[i] == 0)
        i++;

    while (j > i && a[j] != 0)
        j--;

    int temp = a[i];
    a[i] = a[j];
    a[j] = temp;
}
```

A. If $N=1$ and `a` is initialized to be `[1]`

B. If $N=2$ and `a` is initialized to be `[0, 1]`

C. If $N=2$ and `a` is initialized to be `[1, 0]`

D. If $N=200$ and `a` is initialized to contain 100 0's and 100 -9's in random order.

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

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

It moves all the 0's to the beginning of the array

PART 1.

Ahmad, Sami and Ali are twins studying together in the same class. They took this year 20 tests together. Write a complete C program that reads the grades for the three brothers in each of these 20 tests and outputs the number of tests in which Sami scored higher than both of his brothers.

Solution.

```
#include <stdio.h>

int main() {
    int ahmad, sami, ali;
    int count = 0;

    for (int i = 0; i < 20; i++) {
        scanf("%d%d%d", &ahmad, &sami, &ali);
        if (sami > ahmad && sami > ali)
            ++count;
    }

    printf("%d", count);
    return 0;
}
```

PART 2.

Implement a **void** function named **half_doubles(. . .)** that receives an array of distinct positive integers (أرقام مختلفة) and its size as arguments and performs the following:

For every integer in the array, the function prints "Half Found" if the integer's half is also in the array and "Double Found" if the integer's double is also in the array.

The output of the function **must** be in the same **order** and format as the following examples show:

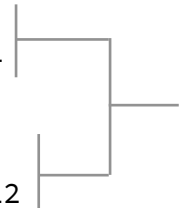
Example 1: a[] = {1, 4, 3}

Output: There should be no output for this example.

Example 2: a[] = {5, 2, 4, 6, 12, 3, 1}

Output:

```
2: Half Found = 1
2: Double Found = 4
4: Half Found = 2
6: Half Found = 3
6: Double Found = 12
12: Half Found = 6
3: Double Found = 6
1: Double Found = 2
```



The half of a number is always printed before the double

Solution.

```
void half_doubles(int a[], int size) {
    for (int i = 0; i < size; i++) {
        for (int j = 0; j < size; j++)
            if (a[j] == a[i] / 2.0)
                printf("%d: Half Found = %d\n", a[i], a[j]);

        for (int j = 0; j < size; j++)
            if (a[j] == a[i] * 2)
                printf("%d: Double Found = %d\n", a[i], a[j]);

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

PART 1. What does the following function return in each of the following cases?

```
int mystery(long n) {
    if (n == 0)
        return 0;

    if (n % 2 == 0)
        return 1 + mystery(n / 10);
    else
        return mystery(n / 10);
}
```

A. If $n = 9$

B. If $n = 10$

C. If $n = 9223372036854775807$

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

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

It counts the number of even digits in n

PART 2. What does the following function return in each of the following cases?

```
int mystery(long n) {
    if (n < 10) return n;

    int d1 = mystery(n / 10);
    int d2 = n % 10;

    if (d1 > d2) return d1;
    else return d2;
}
```

A. If $n = 9$

B. If $n = 10$

C. If $n = 9223372036854775807$

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

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

It returns the maximum digit in n

PART 3.

Implement a **recursive** function named **check** that receives as arguments an array named **a** and two integers named **i** and **j**. The function returns **1** if the array is read the same forward or backward and **0** otherwise.

Examples:

Return **1** for: [1] and [2, 2] and
[3, 7, 4, 4, 7, 3]

Return **0** for: [1, 2, 3] and
[1, 2, 3, 1, 2, 3]

Assume that when the function is called:

i = 0 and **j = the array size - 1**.

Solution.

```
int check(int a[], int i, int j) {
    if (i >= j)
        return 1;

    if (a[i] != a[j])
        return 0;

    return check(a, i+1, j-1);
}
```

PART 1.

Implement a function named **count(...)** that receives as an argument a character array of size **7 x 10**. Each cell in the array is assumed to contain only **'0'** or **'#'**. The function must modify the array such that each **'0'** is changed to become the number of **'#'** around it (above, below, left and right).

Example.

```
0 0 0 # # # # 0 0
# 0 # # 0 0 0 0 # #
0 0 0 0 # 0 0 # # 0
0 # # 0 # 0 0 0 0 0
0 # 0 # 0 0 # 0 # 0
0 0 # # # # # # 0
0 0 # # 0 0 0 # # 0
```

Before calling the function

```
1 0 2 # # # # 2 1
# 2 # # 3 1 1 3 # #
1 1 2 2 # 1 1 # # 2
1 # # 3 # 1 1 1 2 0
1 # 4 # 3 2 * 3 # 1
0 2 # # # # # # 1
0 1 # # 2 1 2 # # 1
```

After calling the function

Solution.

```
#define ROWS 7
#define COLS 10

void count(char a[ROWS][COLS]) {
    for (int i = 0; i < ROWS; i++) {
        for (int j = 0; j < COLS; j++) {
            if (a[i][j] == '*')
                continue;
            if (i != 0 && a[i-1][j] == '#')
                a[i][j]++;
            if (i != ROWS-1 && a[i+1][j] == '#')
                a[i][j]++;
            if (j != 0 && a[i][j-1] == '#')
                a[i][j]++;
            if (j != COLS-1 && a[i][j+1] == '#')
                a[i][j]++;
        }
    }
}
```

PART 2.

Write a complete C program that creates a 2D array of size **7 x 10**, fills it randomly with either '0' or '#' and sends it to function **count(...)**. The program should then print the array after the function call (in the same format as the sample in part 1).

Solution.

```
int main() {
    char a[ROWS][COLS];

    for (int i = 0; i < ROWS; i++) {
        for (int j = 0; j < COLS; j++) {
            if (rand() % 2 == 0)
                a[i][j] = '0';
            else
                a[i][j] = '#';
        }
    }

    count(a);

    for (int i = 0; i < ROWS; i++) {
        for (int j = 0; j < COLS; j++)
            printf("%c ", a[i][j]);

        printf("\n");
    }

    return 0;
}
```

PART 1.

Implement a function named **ends_with(...)** that receives two strings **str1** and **str2** as arguments and returns **1** if **str1** ends with **str2**.

Examples.	str1	str2	result
	HELLO	LLO	1
	HELLO	HELLO	1
	HELLO	OK	0
	HELLO	HELLOO	0
	HELLO	HHELLO	0

Notes:

- The sizes of the strings are not known. Computing them might help in your solution!
- You are **not** allowed to use array notation and/or functions from the `string.h` library in your implementation. Use pointers and pointer arithmetic only.

Solution.

```
int ends_with(char* str1, char* str2) {
    char* ptr1 = str1;
    char* ptr2 = str2;

    int size1;
    for (size1 = 0; *ptr1 != '\0'; size1++)
        ptr1++;

    int size2;
    for (size2 = 0; *ptr2 != '\0'; size2++)
        ptr2++;

    if (size2 > size1)
        return 0;

    for (int i = 0; i < size2; i++) {
        if (*ptr1 != *ptr2)
            return 0;

        ptr1--;
        ptr2--;
    }

    return 1;
}
```


PART 2.

Write a full C program that reads 100 email addresses. For each email, the program must and check if it is a valid PSUT email address (ends with "**@psut.edu.jo**").

Notes:

- You can assume that entered emails do not contain more than **50** characters.
- You *must* call function **ends_with(...)** (from **PART 1**) in your implementation.
- You are not allowed to use any function from the library `string.h`.

Solution.

```
int main() {
    char str[100];
    char suffix[] = "@psut.edu.jo";

    for (int i = 0; i < 100; i++) {
        scanf("%s", str);

        if (ends_with(str, suffix))
            printf("Valid PSUT Email\n");
        else
            printf("Invalid PSUT Email\n");
    }

    return 0;
}
```

This page is intentionally left blank