King Fahd University of Petroleum and Minerals
Information and Computer Science Department
ICS 103: Computer Programming in C

Second Semester 2014 – 2015

Midterm Exam, Thursday April 2, 2015

Time: **120 minutes**

Name: SOLUTION KEY **ID:** _____

Instructor and Section: Select one

Instructor	Section
Dr. Muhamed Mudawar	[ ] **07** (UT 1100) [ ] **09** (UT 1310)
Dr. Samer Arafat	[ ] **17** (MW 0900) [ ] **20** (MW 1100) [ ] **23** (MW 1310)
Dr. Rafi Ul Hasan	[ ] **04** (UT 0800) [ ] **10** (UT 1310)
Dr. Basem Al-Madani	[ ] **01** (UT 0700) [ ] **05** (UT 0800) [ ] **08** (UT 1100)
Dr. Abdulaziz Alkhoraidly	[ ] **02** (UT 0700) [ ] **14** (MW 0800)
Dr. Mohammed Balah	[ ] **03** (UT 0700) [ ] **06** (UT 0800) [ ] **11** (MW 0700) [ ] **16** (MW 0800) [ ] **19** (MW 0900)
Mr. Jaweed Yazdani	[ ] **21** (MW 1100) [ ] **24** (MW 1310)
Mr. Said Muhammad	[ ] **12** (MW 0700) [ ] **15** (MW 0800)
Mr. Hakim Adiche	[ ] **22** (MW 1100)
Mr. Hazem Selmi	[ ] **18** (MW 0900)

Instructions:

1. Answer all questions. Make sure your answers are **clear** and **readable**.
2. The exam is closed book and closed notes. No calculators or any helping aides are allowed. Make sure to turn off your mobile phone and keep it in your pocket.
3. If there is no space on the front of the page, use the back of the page. Indicate this clearly.

Question	Maximum Points	Earned Points	Remarks
1	**15**		
2	**15**		
3	**20**		
4	**15**		
5	**15**		
6	**20**		
Total	**100**		

Question 1 (15 pts): Fill in the blanks

(2 pts) Computer software can be classified into **SYSTEM SOFTWARE (or OPERATING SYSTEM)** and **APPLICATION SOFTWARE**.

(2 pts) A **COMPILER** translates a high-level C program into **MACHINE LANGUAGE (INSTRUCTIONS)**

(2 pts) Specify the correct order for these operations: **Linking, Execution, Translation, Loading**.

First: **Translation**

Second: **Linking**

Third: **Loading**

Fourth: **Execution**

(1 pt) A program must be loaded into the **MAIN MEMORY** before it can be executed.

(2 pts) The secondary storage consists of units, such as **Magnetic Disk** and **Flash Disk**.

(2 pts) A computer has input devices, such as **Keyboard** and **Mouse** (**Camera**, **Microphone**, etc).

(4 pts) The six steps of the software development method are:

Specify the problem

Analyze the problem

Design the algorithm to solve the problem

Implement the algorithm

Test and verify the correctness of the program

Maintain and update the program

Question 2 (15 pts): Expressions

(5 pts) Compute the value of each of the following expressions:

Note: if you write a value without a decimal point then it means an integer value.

Expression	Value
`7.0 + 9 / 2`	`11.0`
`795 % 100 / 10`	`9`
`7 > 5 > 3`	`0`
`(double) 5 / (int) 2.5`	`2.5`
`7 != 4 != 1`	`0`

(6 pts) Write the mathematical expression in the C language. All variables are of type **`double`**.

Mathematical Expression	Equivalent expression in the C language
$\frac{x^2 - 2x}{\sqrt{5 + x^{y-2}}} - y$	`(x*x - 2*x)/sqrt(5 + pow(x, y-2)) - y`
$\frac{\sqrt{\lvert x + y \rvert}}{e^x} + \frac{2}{5}$	`sqrt(fabs(x+y))/exp(x) + 2.0/5.0`

(4 pts) Using DeMorgan's theorem, rewrite the logical expressions by removing the **`!`** that appears outside the parentheses.

Logical Expression	Equivalent logical expression
`!(x < y || x > z)`	`x >= y && x <= z`
`!(ch != 'A' && ch != 'a')`	`ch == 'A' || ch == 'a'`

Question 3 (20 pts): What will be printed by the following program segments?

```
/* 4 points */
printf("This se\nte\\nce is pri\nted o\\n ma\ny li\\nes.");
```

```
This se
te\nce is pri
ted o\n ma
y li\nes.
```

```
/* 6 points */
int x;
double y;
x = 17 / 2 * 2;
y = 6 / 5.0 + 1.999;
printf("**%3.2f**%.4f**%6.1f**\n", y, y+10, y+100);
printf("**%5d**%3d**%2d**", x, x+10, x+100);
```

*	*	3	.	2	0	*	*	1	3	.	1	9	9	0	*	*		1	0	3	.	2	*	*			
*	*				1	6	*	*		2	6	*	*	1	1	6	*	*									

Program Segment	Output

`/* 3 points */`
`int x,y,z;`
 	
`scanf("%d%d%d",&x,&y,&z);`
 	
`if(x%2==0)`
`  if(y%3==0)`
`  if(z%4==0)`
`  printf("A");`
`  else`
`  printf("B");`
`  else`
`  printf("C");`
`else`
`  printf("D");`
 	
`printf("F");`
	Show the output if the user inputs
	`1 2 3`
	
	DF
	
	Show the output if the user inputs
	`6 9 7`
	
	BF

Program Segment	Output

`/* 3 points */`
`int i,j=1;`
 	
`for(i=1; i<=3; i++){`
`  while(j <= i){`
`  printf("A");`
`  j++;`
`  }`
`  printf("B\n");`
`  j = 1;`
`}`
	AB
	AAB
	AAAB

`/* 4 points */`
`void foo(void);`
`int  test(int i);`
 	
`int main(){`
 	
`  char ch;`
`  int n;`
 	
`  scanf("%c",&ch);`
`  while(ch!='n' && ch!='N') {`
`  switch(ch) {`
`  case 'f': foo();`
`  case 'F': printf("F\n");`
`  break;`
`  case 't':`
`  case 'T': n = test(4);`
`  printf("%d\n", n);`
`  break;`
`  default:  printf("D");`
`  }`
`  scanf("%c",&ch);`
`  }`
`  return 0;`
`}`
 	
`void foo(void) {`
`  printf("A");`
`}`
 	
`int test(int i){`
`  return(i*i);`
`}`
	Show the output if the user inputs
	
	`fFtgn`
	
	AF
	
	F
	
	16
	
	D

Question 4 (15 pts): Write a Function that displays a pattern of stars

Write a function (called **stars**) that receives an integer **n** as a parameter and displays a pattern of stars on the screen. The **main** function can call the function **stars** many times and passes to it different values of **n**. Here is an example of the **main** function:

```
#include <stdio.h>

int main(void) {
  stars(2);
  stars(3);
  stars(4);
}
```

The function **stars** displays the following pattern on the screen for different values of the parameter **n**. Write the function **stars**.

Function Call	Output on the Screen
`stars(2);`	`**` ` *`

`stars(3);`	`***`
	` **`
	`  *`

`stars(4);`	`****`
	` ***`
	`  **`
	`   *`

Solution:

```
void stars(int n) {
  int i, j;
  /* Outer loop for counting the lines */
  for (i=0; i<n ; i++) {
    /* Inner loop 1 for printing spaces */
    for (j=0; j<i; j++)
      printf(" ");

    /* Inner loop 2 for printing stars */
    for (j=0; j<n-i; j++)
      printf("*");

    /* Advance cursor to next line */
    printf("\n");
  }
}
```

Question 5 (15 pts): Write a Mathematical Function

The natural logarithm can be approximated by the following series equation:

$$\ln(x) = (x-1) - \frac{1}{2}(x-1)^2 + \frac{1}{3}(x-1)^3 - \frac{1}{4}(x-1)^4 + \cdots$$

Write a function (called **`ln`**) that receives a parameter **`x`** of type **`double`**. It should compute and return as a result the approximate value of **`ln(x)`** using **`20`** terms of the above series. The result of the function **`ln`** should be of type **`double`**. No need to write the **`main`** function.

Solution:

```
double ln(double x) {
  double sum = 0.0;
  double factor = 1.0;
  int i;
  /* repeat 20 times */
  for (i=1; i<=20 ; i++) {
    factor *= (x - 1.0);
    if (i%2 == 1)
      sum += factor / i;
    else
      sum -= factor / i;
  }
  return sum;
}
```

Question 6 (20 pts): Write a Complete Program

Write a program that reads an integer number and calls a function that computes and returns the sum of digits. For example, if the user inputs **6085** then the sum of digits is **6+0+8+5** which is **19**.

You should write two functions:

The **`main`** function should ask the user to input the integer number. If the user enters a positive number then the **`main`** function should call the function **`sum_digits`** to compute and return the sum of digits. If the user enters a negative number then the main function should display an error message and rejects the number. The program should repeat until the user enters **0**. Use **0** to terminate the program. Here is a sample run of the program:

```
Enter a positive integer (or 0 to terminate): 6085
The sum of digits for 6085 is 19
Enter a positive integer (or 0 to terminate): 12456
The sum of digits for 12456 is 18
Enter a positive integer (or 0 to terminate): -417
Error: -417 is a negative number
Enter a positive integer (or 0 to terminate): 0
Program terminated
```

Hint: to calculate the sum of digits, divide by 10 and use the remainder operator %

For example:

```
6085 % 10 = 5 (digit)
6085 / 10 = 608
608 % 10 = 8 (digit)
608 / 10 = 60
60 % 10 = 0 (digit)
60 / 10 = 6
6 % 10 = 6 (digit)
6 / 10 = 0 (stop when zero)
```

Question 6 (cont'd)

Solution:

```
#include <stdio.h>

int sum_digits(int number);

int main() {
  int num;
  int sum;
  do {
    printf("Enter a positive integer (or 0 to terminate): ");
    scanf("%d", &num);
    if (num > 0) {
      sum = sum_digits(num);
      printf("The sum of digits for %d is %d\n", num, sum);
    }
    else if (num < 0)
      printf("Error: %d is a negative number\n", num);
    else
      printf("Program Terminated\n");
  } while (num != 0);
  return 0;
}

int sum_digits(int number) {
  int digit;
  int sum = 0;

  do {
    digit = number % 10;
    number = number / 10;
    sum += digit;
  } while (number !=0);

  return sum;
}
```