KING FAHD UNIVERSITY OF PETROLEUM AND MINERALS

Information and Computer Science Department

2011/2012 Spring Semester (Term 112)

ICS103 Computer Programming in C (2-3-3)

**Midterm Exam**

Monday, 19 March 2012, 06:30 pm

|  |
| --- |
| Student’s Information |
| Name: | Key Solution |
| KFUPM ID: |  |  |  |  |  |  |  |  |  |
| Serial: |  |
| Section: | Abdulla Al-Sukairy | □ 03 (UT 01pm) | □ 10 (UT 10am) | □ 14 (UT 11am) |
| Adil Al-Suhaim | □ 20 (SM 07am) | □ 24 (SM 08am) | □ 28 (SM 11am) |
| Ahmed Al-Mulhem | □ 09 (UT 10am) | □ 13 (UT 11am) | □ 22 (UT 08am) |
| Ali Al-Yousef | □ 17 (UT 07am) | □ 19 (UT 08am) | □ 23 (UT 09am) |
| Amin Al-Hashim | □ 01 (UT 07am) | □ 03 (UT 09am) | □ 07 (UT 08am) |
| El-Sayed El-Alfy | □ 12 (SM 10am) | □ 16 (SM 11am) |  |
| Emad Ramadan | □ 21 (UT 09am) | □ 25 (UT 10am) |  |
| M Balah | □ 18 (UT 07am) |  |  |
| Mohammad Al-Mulhem | □ 04 (SM 09am) | □ 11 (SM 10am) |  |
| Mohammad Felemban | □ 27 (UT 07am) |  |  |
| Nasir Al-Darwish | □ 06 (SM 01pm) | □ 15 (SM 11am) | □ 26 (SM 09am) |
| Rafi Ul Hasan | □ 02 (SM 09am) | □ 08 (SM 01pm) |  |

**IMPORTANT NOTES**

* Fill-in your information above.
* Do NOT start the exam until you are instructed to do so.
* This is a close material exam. So, remove any relevant material.
* Calculators are NOT allowed. If you have one, put it on the ground.
* Mobile phones are NOT allowed. If you have one, switch it off NOW.
* Questions are NOT allowed after the first 20 minutes.
* You are NOT allowed to leave the testing hall at the last 15 minutes. Remain seated and wait for instructions.
* Make sure you have **6** questions and **13** pages including this page.
* Write clearly, briefly, and precisely.

|  |
| --- |
| Scored Marks |
| Question No. | Max. Mark | Score |
| 01 | 20 |  |
| 02 | 20 |  |
| 03 | 20 |  |
| 04 | 10 |  |
| 05 | 10 |  |
| 06 | 20 |  |
| TOTAL | 100 |  |

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

~ Empty Page ~

Question 01 (20 points)

Circle the correct answer. **(2 points each)**

(1) Which of the following are valid identifiers?

 i. R3D3 ii. per-capita iii. phone#

 iv. ice\_cream v. 92\_aardvarks

 a. i, ii, iv, v

 b. i, iv

 c. i, ii

 d. ii, iv, v

 e. All are valid.

(2) Text enclosed in /\* \*/ in a C program \_\_\_\_\_\_\_\_\_.

 a. gives instructions to the processor

 b. declares memory requirements

 c. makes files available

 d. causes a syntax error

 e. is ignored by the C compiler

(3) If num is a variable of type integer and temp is a variable of type double, how could you **correctly** complete this function call?

 scanf("%lf%d", \_\_\_\_\_\_\_\_\_);

 a. num, temp

 b. &num, &temp

 c. temp, num

 d. &temp, &num

 e. none of the above

(4) What would be displayed by the following program? (The symbol '#' stands for one blank character.)

int main(void)

{

 double a, b;

 a = 37.56;

 b = 101.117;

 printf("Is it%6.1f%9.4f", a, b);

 printf("?\n");

 return (0);

}

 a. Is#it##37.6#101.1170?\n

 b. Is#it##37.6#101.1170?

 c. Is#it##37.5#101.1170?

 d. Is#it##37.6#101.117?\n

 e. Is#it##37.6#101.117?

(5) Which one of these is not a name of a C library function?

 a. printf

 b. sqrt

 c. void

 d. scanf

 e. log

(6) What is the result of the following function call?

 pow(2,3)

 a. 8

 b. 9

 c. 8.0

 d. 9.0

 e. none of the above

(7) What will be displayed by the following program?

#include <stdio.h>

void a(void);

void b(void);

int main(void)

{

 a();

 b();

 return 0;

}

void b(void)

{

 printf("ICS103");

}

void a(void)

{

 printf("Hi");

}

 a. HiICS103

 b. ICS103Hi

 c. HiHi

 d. ICS103ICS103

 e. none of the above

(8) If grade is a variable of type double, what will be its value after executing the following statement?

 grade = 5/2;

 a. 2

 b. 2.0

 c. 2.5

 d. 3

 e. none of the above

(9) If sum is a variable of type integer, what will be its value after executing the following statement?

 sum = 84.8;

 a. 84.8

 b. 84.0

 c. 84

 d. 85

 e. none of the above

(10) The equivalent C expression to the following algebraic one is:

$$\frac{x}{y+z}-3$$

 a. x / y + z -3

 b. -3 + x / (y + z)

 c. (x / y + z) - 3

 d. (x – 3) / (y + z)

 e. none of the above

Question 02 (20 points)

**PART 1:** Find the output of the following segments of code.

|  |  |
| --- | --- |
| int i, counter = 0;for(i=0; i<10; i++) counter++;printf(“%d\n”, counter);counter = 0;for(i=15; i<=24; i=i+2) counter++;printf(“%d\n”, counter);counter = 0;for(i=90; i>0; i--) counter++;printf(“%d\n”, counter); | 105901 point each |
| int i = 3, j;do{ for(j=i; j<5; j++) printf("%d\n", i\*j); ++i;} while(i<10); | 9121 point each16 |
| int x, y = 0;scanf(“%d”, &x); // inputswitch(x){ case 1: y = y + 5; break; case 2: y = y + 1; case 3: y = y + 3; default: y = y + 1;}printf(“%d”, y); | **Input: 1**51 point | **Input: 2**51 point | **Input: 5**11 point |
| int x, y;scanf(“%d %d”, &x, &y); //inputif(x > y && y > 5) printf(“First\n”);if(x <= 10 || y < 9) printf(“Second\n”);else printf(“Third”); | **Input: 8 6**FirstSecond | **Input: 6 10**Second | **Input: 12 9**First1 point1 point1 pointThird |

**PART 2:** Find the value of each of the following expressions. Use 1 to indicate true. **(1 point each)**

|  |  |
| --- | --- |
| Expression | Value |
| -10 < -5 < -1 | **0** |
| 3 > 5 && 2 >= 0 | **0** |
| 0 <= 7 || 0 | **1** |
| -1 == 1 == 0 | **1** |
| 5 && !1 && 3 | **0** |
| !1 == 1 | **0** |
| 0 != 1 == 1 | **1** |
| -8 && 0 < 10 | **1** |

Question 03 (20 points)

Fill-in the blanks.

(1) The process of using data files for input/output involves four steps as follows:

a. Declare variables of type \_**FILE**\_ to represent the files. **(2)**

b. Open the files for reading/writing using the \_**fopen**\_ function. **(2)**

c. Read/write from/to the files using the \_**fscanf**\_ and \_**fprintf**\_ functions. **(1 + 1)**

d. Close the files after processing the data using the \_**fclose**\_ function. **(2)**

(2) When one opens a non-existing file for reading, the function \_**fopen**\_ will return \_**NULL**\_.**(2+2)**

(3) The function \_**fscanf**\_ returns the special value \_**EOF**\_, when it reaches the end of a file. **(2+2)**

(4) The function \_**scanf**\_ is used to read data from the keyboard, while the function \_**fscanf**\_ is used to read data from a file. **(2+2)**

Question 04 (10 points)

Complete the following C program that converts a distance from miles to kilometers or from kilometers to miles. Your program should give two options to the user.

* If the user types 1, then the program will ask for a distance in kilometers and converts it to miles.
* If the user types 2, then the program will ask for a distance in miles and converts it to kilometers.
* If the user types any another number, the program will display an error message.

1 mile =1.609 kms. The conversion factor from mile to kilometer needs to be declared as a constant.

Use if-else if statement.

Samples of the program runs are shown below:



 

 

#include <stdio.h>

**#define KMS\_PER\_MILE 1.609 1**

int main(void)

{

**double mile, kms; 0.5**

**int choice; 0.5**

 printf("Enter your choice: 1 (kms to miles) or 2 (miles to kms)> ");

**scanf(“%d”, &choice); 1**

 if (**choice == 1 1**){

 printf("Enter distance in kilometers> ");

**scanf(“%lf”, &kms); 1**

 miles = kms/KMS\_PER\_MILE;

**printf(“%.2f kilometers = %.2f miles\n”, kms, miles); 1**

 }

 else if (**choice == 2 1**){

 printf("Enter distance in miles> ");

**scanf(“%lf”, &miles); 1**

 kms = miles\*KMS\_PER\_MILE;

**printf(“%.2f miles = %.2f kilometers\n”, miles, kms); 1**

 }

 else

**printf(“Sorry! Your choice must be 1 or 2”); 1**

 return 0;

}

Question 05 (10 points)

Complete the following C program that reads unknown number of double grades from a file called **my\_file.txt** and finds and displays the average of all the grades. The program should take care about the case when the file does not exist.

#include <stdio.h>

int main(void)

{

 double grade, sum = 0, avg;

 int count = 0;

**FILE \*inp; 1**

 inp = **fopen(“my\_file.txt”, “r”); 2**

 if(**inp == NULL 1**){

 printf("Error! Can’t open file my\_file.txt\n");

 system("pause");

 exit(1);

 }

 while(**fscanf(inp, “%lf”, &grade) != EOF 2**){

**sum += grade; 1**

**count++; 1**

 }

**avg = sum/count; 1**

**printf(“average = %.2f\n”, avg); 1**

 fclose(inp);

 return 0;

}

Question 06 (20 points)

**PART 1:** Complete the following C program that reads an expression consisting of two integer numbers with a character (between the integers) that represents an operation. This character can be one of the following two symbols: ‘\*’ or ‘+’. Based on this character, the program performs the corresponding operation and prints the result. The program should show an error message if any character other than ‘\*’ or ‘+’ where encountered.  **You must use the switch statement and the output must comply with the format shown in the sample run screens below.**



 

 

#include <stdio.h>

int main(void)

{

 int x, y, result;

 char op;

 printf(“Enter two integers with an operation between them> ”);

 scanf("%d %c %d", &x, &op, &y);

**switch(op) 1**

**{**

 **case ‘\*’: 1**

 **result = x \* y; 1**

 **printf(“%d \* %d = %d\n”, x, y, result); 1**

 **break; 0.5**

 **case ‘+’: 1**

 **result = x + y; 1**

 **printf(“%d + %d = %d\n”, x, y, result); 1**

 **break; 0.5**

 **default: 1**

 **printf(“%c is unknown operation\n”, op); 1**

**}**

 system(“pause”);

 return 0;

}

**PART 2:** Complete the code fragment below in order to convert the following switch statement into **one** logically equivalent if-else structure with **minimum** conditions:

 switch (classID) {

 case 'B':

 case 'b':

 printf("Battleship\n");

 break;

 case 'C':

 case 'c':

 printf("Cruiser\n");

 break;

 default:

 printf("Unknown ship class %c\n", classID);

 }

**if(classID == ‘B’ || classID == ‘b’) 2**

 printf("Battleship\n");

**else if(classID == ‘C’ || classID == ‘c’) 2**

 printf("Cruiser\n");

**else 1**

 printf("Unknown ship class %c\n", classID);

**PART 3:** Complete the following C program that reads an integer number from the user then prints the sum of its digits. For example, if the value read is 1234, the output will be 10 (1+2+3+4). Assume the user will only enter positive values.

Hint: To isolate digits use modulo (%) 10 and divide-by 10 repeatedly.

#include <stdio.h>

int main(void)

{

 int n, digit, sum;

 printf("Enter integer value> ");

 scanf("%d", &n);

**sum = 0 1**

**while(n > 0) 1**

**{**

 **digit = n%10; 1**

 **sum += digit; 1**

 **n /= 10; 1**

**}**

 printf("Sum of digits = %d\n", sum);

 system("pause");

 return 0;

}

~ Scratch Paper ~