2nd SEMESTER UNIVERSITY EXAMINATIONS – 2008/2009

IS 136: C Programming Language

Duration: 2 Hours

Date: 29th July, 2009

Instructions:

1. The examination is divided into Sections A and B.
2. Answer all questions in Section A and any three in Section B.
3. Section A carries 24 Marks and Section B carries 36 marks, giving a total of 60 marks.
4. This paper consists of 2 printed pages.

SECTION A

1. a. Describe the term function as used in C programming. (1.5 Marks)

   b. Write brief notes regarding C functions for each of the following. (4 Marks)
      i. Number of arguments taken and their ordering by data type
      ii. Number of arguments returned
      iii. Changes to values of arguments upon function return
      iv. Status of variables declared within a function upon function return

2. a. Explain the terms prototype and scope as used in C programming. (2 Marks)

   b. Explain what the following code snippet does. Justify your answer. (3 Marks)

```
#include <stdio.h>
void print_stars(int);
int i;

int main(){
        for (i=0; i < 5; i++)  print_stars(5);
        return 0;
}

void print_stars(int n){
        printf("*");
}
```

3. a. The C Language supports the constructs **enum**, **#define** and **const**. Use examples to illustr the syntax for each and state the relative difference between the three constructs. (3 Mark

   b. Distinguish between pass by value and pass by reference. What is the significance of the latter to the **scanf** function? (2 Marks)

4. a. Explain the term pointer and use examples to illustrate the meaning of the operators & and * in relation to pointers. (2 Marks)

   b. Write a function prototype for the function named '**change_argument**' with one argumen changed with the function. Use argument type of your own choice and assume the functio return any value. (3 Marks)

5. a. Explain each of the special streams **stdin**, **stdout**, and **stderr** as defined in the **stdio.h** hea (1.5 Mark

   b. Explain what each of the following code fragment does. (2 Marks)
      i. fprintf (stdout,"Hello World!\n");
      ii. char tLine[50]; fgets (tLine,50,stdin);

SECTION B

a. What is the use of the keywords **struct** and **typedef** in C Language (2 Marks)

b. The C Language does not support a data type line. A line has two coordinates, and each coordinate has X- axis float value and Y-axis float value. Use C structures to define a new data type line, and associate the new type with the name LINE (4 Marks)

c. Write a function named 'is_parallel_lines' that takes two lines as arguments, checks if they are parallel and return 1 if true and 0 otherwise (Hint: two lines are parallel if have same gradient/slope). (6 Marks)

a. What is the purpose of **FILE *** and **fopen** function in relation to file handling in C? (2 Marks)

b. Explain the three modes supported by the **fopen** function. (1.5 Marks)

c. Write C code fragment that performs the following operations:

i. Declare a file pointer 'fptr'.

ii. Declare a character array 'filename' and initialize it with the value "myfile.dat".

iii. Open a file named "myfile.dat" for writing.

iv. If there is failure, a message "Cannot open the file to write!\n" is printed on standard output device, and the program exits with an error code -1. Otherwise, a message "Hello World of filing!\n" is sent to the file "myfile.dat".

v. Close the file "myfile.dat" (6 Marks)

d. Suppose we want to open the file "myfile.dat" for reading, and print every line to the standard output device. Using **fgets** function and a suitable loop, write a code fragment that performs the preceding operations. (2.5 Marks)

a. Distinguish between static(stack) and dynamic(heap) memory as used in the C language (2 Marks)

b. Explain the C memory management functions **malloc**, **free**, **calloc**, and **realloc** (4 marks)

c. Suppose we want to request memory dynamically enough to store 1000 float values, print to the standard error device a message "Out of memory!\n" followed by a call to the exit function with code -1, if there is a failure. Otherwise, put 1000 floats values in subsequent memory slots to be computed by the formula PI * i*i, where PI = 3.14, and i runs from 1 to 1000. In turn, print all values put in the memory block to the standard output device and release the allocated memory block. Write C code fragment that perform all preceding operations using **calloc**. (6 Marks)

State three differences between an array and a pointer. (3 Marks)

State any error in the following code snippets: (4 Marks)

i.
```
int *a;  *a= 3;
```

ii.
```
int *a; a= malloc (100*sizeof(int)); a[100]= 3;
```

iii.
```
int *a;
a= malloc (100*sizeof(int));
...
free (a);
...
*a= 3;
```

Supported by example code fragments briefly explain the following in relation to C pointers:

i. Rogue pointers

ii. Memory leaks (5 Marks)

State any three classic user input errors. (3 Marks)

What is buffer overflow? Discuss how evil/malicious programs exploit buffer overflow. (4 Marks)

What is a wrappered function useful for? (2 Marks)

Given function prototypes *int (*pfi) (int, int);* *int f1();* and int *f2(int, int)*; with reasons state whether each of the following is a valid statement: (3 Marks)

i. *(*pfi)(1,2) ;*

ii. *pfi = f1;*

iii. *pfi = f2;*