UNIVERSITY OF WINDSOR
COMPUTER SCIENCE 60-140-01
MIDTERM   TEST SOLUTION

Examiners: Dr. C.I. Ezeife   Given: Fri., Nov. 7, 2008

Student Name:_____

Student Number:_____

Lab. Section(circle one): 51, 52, 53, 54, other _____
_____

INSTRUCTIONS (Please Read Carefully)

Examination Period is 1 hour 30 minutes. DO NOT WRITE WITH PENCIL.
Answer all questions. Write your answers in the spaces provided in the question paper.
May use only simple calculators.
Total Marks = 100.   Total number of questions = 3. Total number of pages = 7.

_____
For Marking Purposes Only (this part not to be filled by students)

Question	Mark
1 (35 marks)	
2 (15 marks)	
3a (20 marks)	
3b (10 marks)	
3c (20 marks)	
Total Mark (100 marks)	

Problem 1:

1. Write a C program (using only top-down design approach with functions FindProd and FindSum as given in the structure chart below), to compute and print the name and amount of money spent by Tom and Jane individually and together during their recent pre-Christmas shopping. During the shopping, they both bought different quantities of music CDs costing $10.00 each, and computer memory keys costing $20.00 each. …………(35 marks)

DO NOT USE GLOBAL VARIABLES. You should make only parameter calls and read all input data as well as print all desired output data in the main function. All arithmetic operations can be performed only in the functions FindProd or FindSum and not in main.
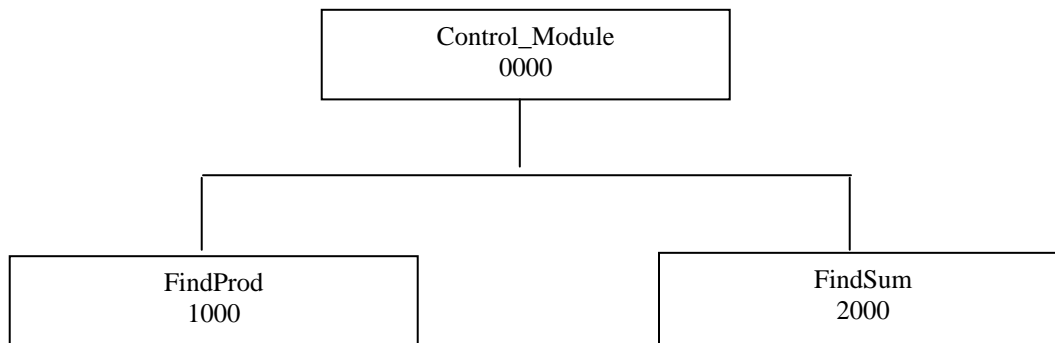
Sample Input:

Type Name1	Number of CDs	Number of Memory Keys:
Tom	5	2
Type Name2	Number of CDs	Number of Memory Keys:
Jane	4	3

Sample Output:
Tom   spent $90.00
Jane   spent $100.00
They both spent $190.00

******** (note that this line is not part of output of programs ********

Use the following structure chart for solving the problem.

```
          ┌─────────────────┐
          │ Control_Module  │
          │      0000       │
          └─────────────────┘
                  │
        ┌─────────┴─────────┐
  ┌───────────┐       ┌───────────┐
  │ FindProd  │       │  FindSum  │
  │   1000    │       │   2000    │
  └───────────┘       └───────────┘
```

SOLUTION

```c
#include <stdio.h>
float FindProd(float, float);
float FindSum(float,float);

void main(void)
{
     char Name1[10], Name2[10];
     int NumCD1, NumCD2, Numkey1, Numkey2;
     float Tomamt, Janeamt, Totalamt;

/* now we present the body of the main program  */

    printf("Type Name1      Number of CDs      Number of Memory Keys:\n");
    scanf("%s  %d  %d", Name1, &NumCD1, &Numkey1);
    printf("Type Name2      Number of CDs      Number of Memory Keys:\n");
    scanf("%s  %d  %d", Name2, &NumCD2, &Numkey2);
    Tomamt = FindSum(FindProd(NumCD1, 10.00), FindProd(Numkey1, 20.00));
    Janeamt = FindSum(FindProd(NumCD2, 10.00), FindProd(Numkey2, 20.00));
    Totalamt = FindSum(Tomamt, Janeamt);
     printf("%s spent %0.2f \n", Name1, Tomamt);
     printf("%s spent %0.2f \n", Name2, Janeamt);
     printf("They both spent %0.2f \n", Totalamt);
}

/* The function definitions for FindProd and FindSum are  next */

float FindProd(float Num1, float Num2)
{
     return (Num1 * Num2);
}

float FindSum(float Num1, float Num2)
{
      int Sum;
      Sum = Num1 + Num2;
      return (Sum);
}
```

Marking Scheme:

Assign -   5 marks for proper definition of input and output variables/constants
-     5 marks for proper declaration of function prototypes
-     20 marks for correct logic in the control module body including correct use of program instructions and function calls.
     (broken down as 5 marks for correct reading of input data, 10 marks for correct function calls, 5 marks for correct printing of output data and messages)
-     5 marks for function definition (2 marks for correct logic and 3 marks for correct function header and parameter passing)
***** the above will make up 35 marks and marks are lost for the following:
-     10 marks off for improper parameter passing mechanisms.
-     Take 15 marks off if parameters are not used in the solution. Then, mark the solution based on 2 marks for variable declaration, 3 for function prototypes, 5 marks for control module body, 5 marks for function definition and 5 marks for correct printing of output.
-     Take 20 marks off for not using functions at all. Then, mark the solution based on 5 marks for variable declaration, 5 marks for control module body, and 5 marks for correct printing of output.
-     10 marks off for using a different structure chart.

2.   This question wants you to trace (that is, execute with hand the way the computer would) through the following simple program (with no functions) and fill in the output of the program in the chart provided below for this question. Show how you arrived at your result for full marks. To show your work, you can show by the side the values of all variables in memory as each instruction is executed.   (15 marks)

```c
#include  <stdio.h>
void main(void)
{
    /* Declaration of variables  */
    int knt=20, num = 4, flag=1, newnum;
    float   cash=54.37;
    char   Initial='M', firstname[10] = "John";

    knt *= 5;
    newnum = knt %  num + knt / num – 1;
    printf("knt is %d\n", knt);
    printf("newnum is %d \n", newnum);
    printf("%s has $%d\n", firstname, (int)cash % 10);
}
```

The output printed by the above program's 3 printf are:

Which printf	Output by Program's printf is to be written on this column
First printf	knt is 100
2ⁿᵈ printf	newnum is 24
3ʳᵈ printf	John has $4

Marking Scheme:
Assign –5 marks for each correct tracing of changes and printing of correct value by the printf for a total of 15 marks for the three printf. We allowed a mark of only 2 for each correct tracing with wrong calculation.

3. This question wants you to trace (execute with hand) through a program that has functions and uses call-by-reference, call-by-value as well as some global and local variables. It also wants you to identify important parts of this program as described later.

Given the following program solution to a problem, answer questions 3a to 3c using this solution. Each of the 10 answers in question 3a to 3c is worth 5 marks.

(50 marks total)

```c
#include  <stdio.h>
/* Global variables  */
int Num4, Num5, Num6, Sum, Diff;

/* Function Prototypes are declared now */
void FindDiff(int, int, int *, int *);
int FindSum(int, int, int);

void main(void)
{
    int  Num1, Num2, Num3, Diff1, Sum1;

/* Executable instructions in main follow  */
    scanf("%d %d %d", &Num1, &Num2, &Num3);
    scanf("%d %d %d", &Num4, &Num5, &Num6);
    FindDiff(Num1, Num2, &Diff1, &Sum1);
    printf("%d  %d \n", Diff1, Sum1);
    FindDiff(Num4, Num6, &Diff1, &Sum1);
    printf("%d  %d\n", Diff1, Sum1);
    Sum = FindSum(Num3, Num5, Num1);
```

```
        printf("%d %d\n", Sum, Diff);
}
int Num7 = 10;
/*  The function definitions are presented next  */
void FindDiff(int first, int second, int *diffp, int *sump)
{
      *diffp = (first - second) / 2;
      *sump = (first + second) + (*diffp);
      printf("  %d  %d\n", *diffp, Num7);
}
int FindSum(int Num1, int Num2, int Num3)
{
   int  sums;
      sums = Num1 + Num2 + Num3;
      Diff = sums * 10;
      return(sums);
}
```

3a.)   With a sample test data as follows: the values 10 5 2 for Num1, Num2, Num3
are typed at the keyboard respectively, then 3 2 1 at the keyboard for Num4, Num5 and
Num6 respectively, write the values printed by the three printf instructions in main and
one printf instruction in FindDiff function in the table below. Show your work to get full
marks here. To show your work, you can show by the side the values of all variables in
memory as each instruction is executed.

After printf instruction at Position..	Values Printed by CPU in correct order are
First printf in main	2 17
Second printf in main	1 5
Third printf in main	14 140
Show all values ever printed by the printf in FindDiff in order	2 10 1 10

3b.) List all global and local variables of the functions in the table below:

Function	Global variables	Local variables
main	Num4, Num5, Num6, Sum, Diff	Num1, Num2, Num3, Diff1, Sum1
FindSum	Num4, Num5, Num6, Sum, Diff, Num7	sums

3c.) Fill in the correct description of listed parts of the program solution of question 3 in the table below: (Place your answer to each question in the column on its right).

A function call in the main is ..	FindDiff(Num1, Num2, &Diff1, &Sum1); or FindDiff(Num4, Num6, &Diff1, &Sum1); or Sum = FindSum(Num3, Num5, Num1);
A list of formal parameters in the program is. Provide the complete list with the brackets..	(int first, int second, int *diffp, int *sump) used in function definition for FindDiff. Another formal parameter list is: (int Num1, int Num2, int Num3) used in the function definition for FindSum
A list of actual parameters in the program is. Provide the complete list with the brackets..	(Num1, Num2, &Diff1, &Sum1) or (Num4, Num6, &Diff1, &Sum1) used in making a function calls to FindDiff. Another actual parameter list is: (Num3, Num5, Num1) used in making a function call to FindSum
A function prototype in the program of Problem 3 is ..	void FindDiff(int, int, int *, int *); or int FindSum(int, int, int);

Marking Scheme:
Take 5 marks off for any of the 10 questions in 3a to 3c incorrectly answered. For question 3a, we allow only a maximum mark of 1 (that is – 1.5) for each correct tracing or correct calculation with written answer wrong. For question 3b, we deduct 0.5 mark for each variable in the list missed or for each extra wrong variable included.