

**University of Windsor (solution)**  
**COMPUTER SCIENCE 60-140-01**  
**FALL FINAL EXAMINATION**

<i>Name:</i>	
<i>ID#:</i>	
<i>Section:</i>	<b>01</b>
<i>Lab Section:</i>	<b>51 52 53 54</b>

**Examiners: Dr. C. I. Ezeife**  
**Given: Friday, Dec. 19, 2008**

---

**PLEASE READ CAREFULLY**

- Examination Period is 3 hours.
- Section A is worth 70% and Section B is 30%.
- Answer all questions.
- Write your answers in the spaces provided in the question paper

<i>Question</i>	<i>Mark</i>
<b><i>Section A</i></b>	
<b><i>1</i></b>	<b><i>/10</i></b>
<b><i>2</i></b>	<b><i>/10</i></b>
<b><i>3</i></b>	<b><i>/10</i></b>
<b><i>4</i></b>	<b><i>/10</i></b>
<b><i>5</i></b>	<b><i>/10</i></b>
<b><i>6</i></b>	<b><i>/10</i></b>
<b><i>7</i></b>	<b><i>/10</i></b>
<b><i>Section B</i></b>	
<b><i>1</i></b>	<b><i>/30</i></b>
<b><i>Total</i></b>	<b><i>/100</i></b>

## Section A (70 marks)

This section has 7 short answers or program questions. Each question in this section is worth 10 marks.

1. Write a program that: Reads in the values of a one dimensional array of 10 integers called Num and calls a function *Large* that would return the largest element of the array. Your program should print the largest element from the main function. If the sample input is the list:  
10 43 67 15 2 60 45 89 90 38

*The output should be:*

*The largest element of array Num is:                    90*

```
#include <stdio.h>

int Large(int [], int);
void main()
{
    int Num[10];
    int r;

    for (r=0; r<10; r++)
    {
        printf("Please scan in element %d of Array Num: ", r);
        scanf("%d", &Num[r]);
    }
    printf("\nThe largest element of array Num is %d\n", Large(Num,
10));
}

int Large(int Num[], int n)
{
    int i, large;
    large = Num[0];
    for (i=1; i<n; i++)
        if (Num[i] > large)
            large = Num[i];
    return(large);
}
```

Marking Scheme: Assign 2 marks for variable declarations in main, 2 marks for proper reading of the array Num, 2 marks for declaring and 2 marks for calling the function Large well, 2 marks for printing the result.

2.

- a. Given the following program segment, what is the equivalent “*Negative Nested if-logic form*”? Recall that Negative if-logic has the sequence “if ...if ...else ..else” as opposed to the positive if-logic sequence of “if ...else if ...else if ....”.

```
if      (temp > 7)
    printf(“I will probably go out for a walk\n”);
else if (temp > 0)
    printf(“I would rather stay home \n”);
else
    printf(“I will definitely stay home \n”);
```

<i>Your answer</i>	<pre><b>If (temp &lt;= 7)     If (temp &lt;=0)         printf(“I will definitely stay home \n”);     else         printf(“I would rather stay home \n”); else     printf(“I will probably go out for a walk\n”);</b></pre>
------------------------	--

**Marking Scheme: 2 marks for correct outer if condition, 2 marks for correct inner if condition and 1 mark for the rest of the instruction.**

- b. Trace the following program and write the output of program for n = 12?

```
#include <stdio.h>

int guess (int);

int main(void)
{
    int i, n;
    int sum;
    printf("Please scan in a positive integer:\t");
    scanf("%d", &n);
    sum = guess(n);
    printf("\nThe value returned by Guess is %d\n", sum);

    return(0);
}

int guess (int n)
{
    int i, sum=0;
    for (i=2; i<=n; i+=2)
        sum+= i;
    return(sum);
}
```

}

<b><i>Your answer</i></b>	Please scan in a positive integer: 12 The value returned by Guess is 42
-------------------------------	--

**Marking Scheme: Assign 5 marks for correct tracing and result. Assign 2 marks if they showed their work but missed the result due to some errors.**

3. Write a program that would:

- Read the following data from a disk file called “*input.txt*”, increment each data value read by 3 and print resulting table of values to the monitor. Solve using double nested loop instructions.

Sample Input is:

5	7	9	11
10	14	18	22
15	21	27	33
20	28	36	44

**Sample Output:**

The new values of table are:

8	10	12	14
13	17	21	25
18	24	30	36
23	31	39	47

```
#include <stdio.h>

int main(void)
{
    FILE *p;
    int A[4][4], i, j;
    int sum;

    p = fopen("input.txt", "r");

    for (i =0; i<4; i++)
        for (j=0; j<4; j++)
            fscanf(p, "%d", &A[i][j]);
    fclose(p);
    for (i =0; i<4; i++)
        for (j=0; j<4; j++)
            A[i][j]+=3;

    printf("The new values of table are:\n");
    for (i =0; i<4; i++)
    {
        for (j=0; j<4; j++)
            printf("%2d\t", A[i][j]);
        printf("\n");
    }
    return(0);
}
```

Marking scheme: Assign 3 marks for correct input/output and file pointer variable declarations, 2 marks for correctly opening the file for read, 2 marks for correct repetition logic, 2 marks for correctly reading from the file and 1 marks for closing the file.

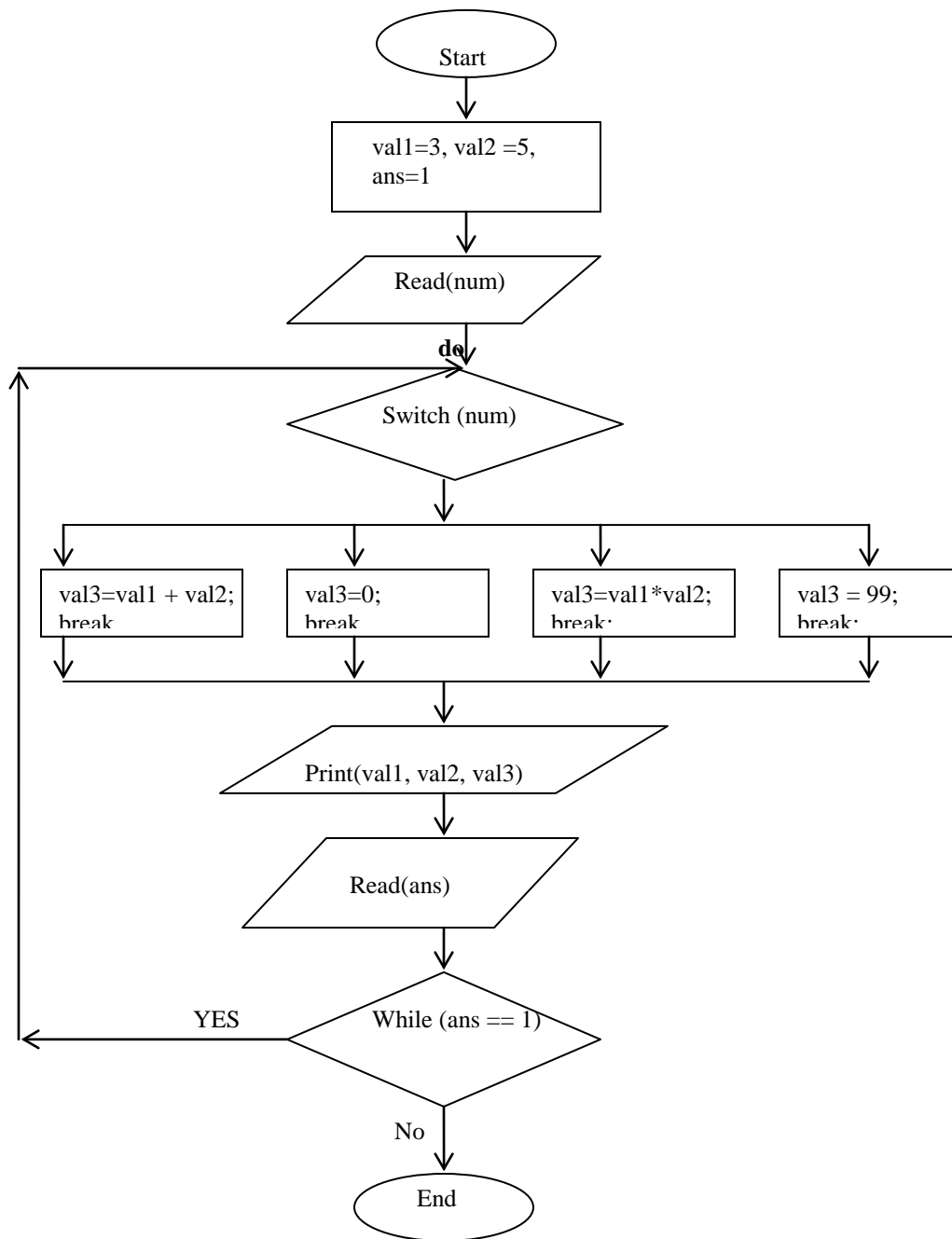
**4. Draw a flowchart for the following program:**

```
#include <stdio.h>
void main(void)
{
    int num;
    int val1=3, val2 =5, val3;
    int ans=1;
    do
    {
        printf("Please scan in an integer:\t");
        scanf("%d", &num);
        switch (num)
        {
            case 1:
                val3=val1 + val2;
                break;
            case 2:
                val3=0;
                break;
            case 3: val3= val1 * val2;
                break;
            default :
                val3 = 99;
        }
        printf ("Val1= %d\nVal2= %d\nVal3= %d\n", val1, val2, val3);
        printf("Please type in 1 if to scan another integer:\t");
        scanf("%d", &ans);
    }
    while (ans ==1);
}
```

**Solution**

\*\*\*\*\*

Given below.





**Marking Scheme: Assign 2 marks for getting the initialization part correct, 2 marks do\_while part, 3 marks for the switch\_case part and 2 marks for updating inside the loop and 1 mark for every thing else.**

5. Find the 3 errors in the following program and correct them, marking your answer on the right of the line of the code that has an error. This program is supposed to scan several values of age and find the smallest value scanned. The value 0 assigned to the variable age is the sentinel value.

```
#include <stdio.h>

void main(void)
{
    float Min;
    float age;

    scanf ("%f", &age);
    Min = age;
    while (age == 0)                                **      while (age !=0)
    {
        if (age < Min)
            age = Min;                             **      Min = age;

                                                    <=inser here**scanf("%f", &age);
    }
    printf ("Min is %.2f\n", Min);
}
```

**Marking Scheme: Assign 3 marks each for each of the found logical errors and 1 bonus mark if they got at least 2 of the errors correct.**

6. Write a program that prints the following shape using only nested repetition instructions. You can use printf statements to print asterisk character '\*'.

```
*
**
***
****
*****
```

```
#include <stdio.h>

int main()
{
    int r, c;
    for (r=1; r<=5; r++)
    {
        for (c=0; c<r; c++)
            printf("*");
        printf("\n");
    }
    return(0);
}
```

Marking Scheme: Give 4 marks for computing the right outer loop and 4 marks for right inner loop and 2 marks for correct printing. Ignore if they forget to print newline.

7. Given three student names in the order lastname firstname, write a C program to read these names and store them in an array and print them nicely in a table.

```
#include <stdio.h>
#include <string.h>

void main(void)
{
    /* The simple program that reads three student names
       as lastname firstname, and stores them in an array to print
       them nicely in a table */

    int r;
    char  firstname[3][10], lastname[3][10];
    for (r= 0; r< 3; r++)
    {
        printf("please, type the %d lastname firstname of input\n", r);
        scanf("%s %s", lastname[r], firstname[r]);
    }
    printf("FirstName \t LastName \n");
    printf("*****\n");

    /* Now print the table of results */
    for (r=0; r<3; r++)
    {
        printf("%s \t \t %s \n", firstname[r], lastname[r]);
    }
}
```

**Marking Scheme: Assign 3 marks for correct declaration of string array, 3 marks for correct reading of data into the string array and 3 marks for correct printing of results and 1 mark for remembering to include string.h.**

**Section B (30 marks)**

**This section has 1 full programming question.**

**Use arrays, functions with parameters (NO global variables)**

1. A class has 4 students and there are 5 assignments for deciding each student's grade in a course. You are required to write a program to calculate the students' grades based on the highest score in the class given that:

Grade

- A will range from the highest score, to the highest score minus 3.0.
- B from the highest score minus 3.1, to the highest score minus 5.0.
- C from the highest score minus 5.1, to the highest score minus 7.0.
- D from the highest score minus 7.1, to the highest score minus 9.0.
- F will be anything below highest score minus 9.0.

Your program should read each student's name as a string, the student's assignment marks. The output will be each student's name, assignment marks, average assignment score, and grade, along with the single highest average score for the class. You should use the structure chart given below for solving the problem. Note that the function ReadD is for reading all input data, the function FindGrade is for computing the average marks, each student grade and the highest average mark, while the function PrintData is for printing the entire table of output values as shown in the sample output below.

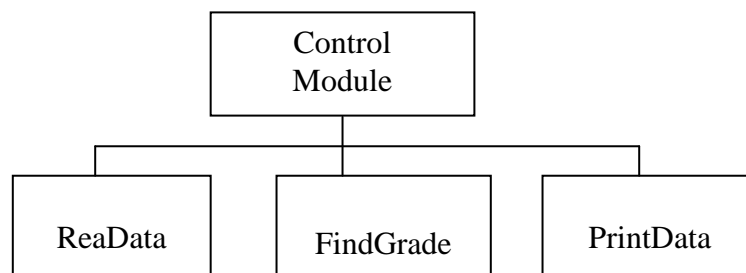
**Sample Input Data**

John_Smith	70	51	65	83	85
Mary_Kay	83	65	67	71	91
Peter_Good	90	70	71	77	80
Ryan_Oops	92	85	88	80	79

**Sample Output Data is:**

Student Name	Assignment Marks					Student Average	Grade
John_Smith	70	51	65	83	85	70.8	F
Mary_Kay	83	65	67	71	91	75.4	F
Peter_Good	90	70	71	77	80	77.6	D
Ryan_Oops	92	85	88	80	79	84.8	A

Highest Average Score = 84.8



## Solution

---

```
/*To be able to test the program,
 *Calculate grades based on the highest score in the class
 *A: from the highest score to the highest score-3
 *B: from the highest score-3.1 to the highest score-5.0
 *C: from the highest score-5.1 to the highest score-7.0
 *D: from the highest score-7.1 to the highest score-9.0
 *F: below the highest score-9.0
 */

#include <stdio.h>

/*declare function prototypes*/
void ReadD (char [][][15], int [][][5]);
float FindGrade (int [][][5], float [], char []);
void PrintData(char [][][15], int [][][5], float [], char [], float);

void main () {
    char Name[4][15], grade[4];
    int marks[4][5];
    float average[4], highestavg;

    ReadD (Name, marks);
    highestavg = FindGrade (marks, average, grade);
    PrintData(Name, marks, average, grade, highestavg);
}

/*read data into array of studentID, answers, answer key*/
void ReadD (char Name[][15], int marks[][5])
{
    int r, c;
    for (r=0; r<4; r++) {
        printf("Type student name and hit enter\n");
        scanf ("%s", Name[r]);
        printf("Type the student 5 marks in order\n");
        for (c=0; c<5; c++){
            scanf ("%d", &marks[r][c]);
        }
    }
}

/*calculate grade of each student*/
float FindGrade (int marks[][5], float average[], char grade[])
{
    int r, c, bestsc, rowsum[4]={0,0,0,0};
    float bestsc = 0.0;

    /*find highest score*/
    for (r=0; r<4; r=r+1) {
        for (c=0; c<5; c=c+1)
        {
            rowsum[r] += marks[r][c];
        }
        average[r] = (float) rowsum[r]/5.0;

        if (average[r] > bestsc) bestsc = average[r];
    }

    /*calculate grade of each student*/
```

```

for (r=0; r<5; r=r++) {
    if (average[r] >= bestsc-3.0)
        grade[r] = 'A';
    else if (average[r] >= bestsc-5.0)
        grade[r] = 'B';
    else if (average[r] >= bestsc-7.0)
        grade[r] = 'C';
    else if (average[r] >= bestsc-9.0)
        grade[r] = 'D';
    else
        grade[r] = 'F';
}
return bestsc;    /*return the highest score*/
}

/*print result*/
void PrintData (char Name[][15], int marks[][5], float average [], char
grade[], float bestsc)
{
    int r, c;
    printf ("Student Name \t Assignment Marks\t\t Student Average \t
Grade\n");
    for (r=0; r<4; r++){
        printf ("%s\t ", Name[r]);
        for (c = 0; c < 5; c++)
            {
                printf("%d\t", marks[r][c]);
            }
        printf("%0.2f\t \t %c \n", average[r], grade[r]);
    }

    printf("Highest Average Score = %0.2f\n", bestsc);
}

```

#### Marking Scheme:

Proper declaration of all input and output variables	worth	4 marks
Proper declaration of function prototypes	worth	3 marks
Proper logical sequencing and correct definition of control module body		
Including correct parameter calls to functions.		9 marks
Proper definitions of the function headers with parameters		3 marks
Logical correctness of function definitions body		9 marks
Correct printing of output and headers		2 marks
		-----
Total		30 marks

Note: If a student fails to use functions and parameters, they lose nearly all marks for these components if their logic is correct.