THE UNIVERSITY OF MELBOURNE
DEPARTMENT OF COMPUTING AND INFORMATION SYSTEMS

SAMPLE FINAL EXAM TWO, SOLUTION – SEMESTER 2, 2016
COMP20005 ENGINEERING COMPUTATION

Student ID:

Reading Time: **fifteen** minutes
Writing Time: **two** hours

Total marks for this Exam: 60
This exam has 4 pages.

Identical Examination Papers: None
Common Content Papers: None

Authorised Materials:
Writing materials, e.g., pens, pencils, are allowed.
Books, calculators, and dictionaries are *not* allowed.

Instructions to Invigilators:
Supply students with standard script book(s).

The exam paper must remain in the exam room and be returned to the subject coordinator.

Instructions to Students:

- Attempt all questions.
- You may attempt the questions in any order. *However, you should write your answers that belong to the same question together.*
- Clearly write your answers. Any unreadable answer will be considered wrong.
- You are not required to write comments in any of your code fragments or functions. If a question says "write a function", you may write appropriate further functions if you believe that a decomposition of the problem is appropriate.
- You may make use of library functions except when their use is explicitly prohibited. If you do make use of library functions, you must add suitable `#include` lines at the start of each corresponding answer.
- Constants should be `#define`'d prior to functions, when appropriate.

1. Short answer questions **[3 marks for each question]**

(1) Assume a system where int variables are represented by 8-bit two's complement number representation. If an int variable var = 127, then what is the corresponding bit representation of var? What is the value of var (in decimal representation) after we increase var by 1 (that is, var = var + 1)?

01111111

-128

(2) Let $f(x) = x^2 - 10$. Write out the values of x_1, x_2, and x_m for the first 3 iterations to find the root of $f(x) = 0$ using the bisection method, starting at $x_1 = 0$ and $x_2 = 10$.

x_1	x_2	x_m	f(x_1)	f(x_2)	f(x_m)
0	10	5	-10	90	15
0	5	2.5	-10	15	-3.75
2.5	5	3.75	-3.75	15	4.0625

(3) State the possible problems of applying the Euler forward difference approach if either too large or too small values are used for Δt.

Make Δt too big, then the over- or under-shoot that occurs at every step can quickly become divergent.

But make Δt too small, then rounding/truncation errors can become a risk.

(4) Name two methods that can be used to calculate numeric integration.

Trapezoidal method, Simpson's method

(5) Describe in no more than five English sentences the procedure of determining whether a given integer is in a sorted array of integers using the divide and conquer strategy.

1. check the middle integer in the array
2. if the middle integer is the given integer, return 1
3. if the middle integer is larger, recursively check (only) the first half
4. if the middle integer is smaller, recursively check (only) the second half
5. if the given integer is not found in either half, return 0

Programming questions

2. **[10 marks]** Write a function int countPerfectNumber(int array[], int n) that calculates and returns the number of "*perfect numbers*" in the array of n (n > 0) positive integers. Here, a perfect number is a number that equals to the sum of its factors (including the factor 1, but excluding the number itself, and 1 is not a perfect number).

   You need to add suitable `#include` lines if you use any library functions.

```
int countPerfectNumber(int array[], int n){
	int i;
	int count = 0;

	for (i = 0; i < n; i++) {
		count += isPerfect(array[i]);
	}

	return count;
}

int isPerfect(int num) {
	int i, factorSum = 0;

	if (num == 1) {
		return 0;
	}
	for (i = 1; i < num; i++) {
		if (num % i == 0) {
			factorSum += i;
		}
	}
	return factorSum == num;
}
```

3. **[10 marks]** Write a function int str2Int(char *str) that converts a string str into the corresponding integer and returns the integer. For example, if str = “123”, then the function should return 123.

   You may assume that str is always valid (that is, can be converted into an integer). You may **NOT** make use of any functions in the <string.h> or <stdlib.h> libraries.

```
int str2Int(char *str) {
        int n = 0;
        int i = 0;
        int sign = 1;
        if (str[0] == ‘-’) {
                sign = -1;
                i = 1;
        }
        while (str[i]) {
                n = n*10 + str[i]-‘0’;
                i++;
        }

        return n*sign;
}
```

4. [15 marks] In COMP20005, a student's record consists of a unique student ID represented by an integer, four marks (for mid-semester exam, Assignment 1, Assignment 2, and the final exam) represented by four floating point numbers, and a grade represented by a character.

(1) Write out the definition of a struct type named student_t according to the description above.

(2) Assume that you are given a student_t typed array named students which contains the records of n students (n > 0). Each of these records has already got a student ID and the four marks, but the grade field does not have a value yet.

Write a function void fillGrade(student_t students[], int n) that fill in the grade for each student according to the following rules.

If the student has failed the subject, then the grade should be 'F'. Note that a student is considered failed if his/her total mark is less than 50, or his/her total exam mark is less than 28, or his/her total assignment mark is less than 12.

If the student has a total mark of 80 or above, then the grade should be 'H'.

All other students should be graded 'P'.

(3) Write another function void sortRecordsByGrade(student_t students[], int n) that sorts the students array in the descending order of the grades, where 'H' is deemed greater than 'P', and 'P' is deemed greater than 'F'. If there are two student records with the same grade, then the record with a smaller student ID should be put in the front.

You need to add suitable #include lines if you use any library functions.

```
#define NUM_MARKS 4
#define MID_EXAM 0
#define FINAL_EXAM 1
#define ASSMT_1 2
#define ASSMT_2 3
#define FAIL 50
#define EXAM_HURDLE 28
#define ASSMT_HURDLE 12

typedef struct {
        int id;
        double marks[NUM_MARKS]; /* can use float as well */
        char grade;
} student_t;
```

```
void fillGrade(student_t students[], int n) {
        int i;
        double examTotal, assmtTotal, overallTotal;

        for (i = 0; i < n; i++) {
                examTotal = students[i].marks[MID_EXAM] +
                        students[i].marks[FINAL_EXAM];
                assmtTotal = students[i].marks[ASSMT_1] +
                        students[i].marks[ASSMT_2];
                overallTotal = examTotal + assmtTotal;

                if (examTotal < EXAM_HURDLE ||
                        assmtTotal < ASSMT_HURDLE ||
                        overallTotal < FAIL) {
                        students[i].grade = 'F';
                } else if (overallTotal >= 80) {
                        students[i].grade = 'H';
                } else {
                        students[i].grade = 'P';
                }
        }
}
int isGreater(student_t student1, student_t student2) {
        if (student1.grade == student2.grade) {
                return student1.id < student2.id;
        }
        If (student1.grade == 'H' || (student1.grade == 'P' && student2.grade == 'F')) {
                return 1;
        }
        return 0;
}
void swap(student_t *student1, student_t *student2) {
        student_t tmp;
        tmp = *student1;
        *student1 = *student2;
        *student2 = tmp;
}
void sortRecordsByGrade(student_t students[], int n) {
    int i, j;
    for (i=1; i<n; i++) {
        for (j=i-1; j>=0 && isGreater(A[j+1], A[j]); j--) {
                swap(&students[j], &students[j+1]);
        }
    }
}
```

5. **[5 marks]** Write a ***recursive*** function int fibonacci(int n) that calculates and returns the *Fibonacci number* F(n), where

$$F(n) = \begin{cases} F(n-1) + F(n-2), & n > 2 \\ 1, & n = 1, 2 \end{cases}$$

You may assume n > 0.

If you use iteration rather than recursion to answer this question, the full mark of this question will reduce to **2 marks**.

```
int fibonacci(int n) {
        if (n == 1 || n == 2) {
                return 1;
        }
        return fibonacci(n-1) + fibonacci(n-2);
}

// Iterative solution
int fibonacci(int n) {
        int i = 3;
        int nextFibo, fibo1 = 1, fibo2 = 1;
        if (n == 1 || n == 2) {
                return 1;
        }
        while (i <= n) {
                nextFibo = fibo1 + fibo2;
                fibo1 = fibo2;
                fibo2 = nextFibo;
                i++;
        }

        return nextFibo;
}
```

6. **[5 marks]** When evaluating mathematics expressions, it is important to make sure that the parentheses are correctly placed. Given a mathematics expression, we say that the parentheses are correctly placed if the number of opening parentheses and the number of closing parentheses are the same, and that within any prefix of the expression, the number of opening parentheses is not less than the number of closing parentheses. For example, “1+(2*3)” is a correct use of parentheses, while neither “(1+2*3” nor “1)+(2*3” is a correct use.

   Your task is to write a function int isParenthesesCorrect(char *exp) that checks a mathematics expression stored in a string exp, and returns if the parentheses are correctly used in exp (1 for yes and 0 for no).

   You may assume that exp is not NULL. You may **NOT** make use of any functions in the <string.h> library. You need to add suitable `#include` lines if you use any other library functions.

```
int isParenthesesCorrect(char *exp) {
        int i = 0, openingCount = 0;

        while (exp[i]) {
                if(exp[i] == '(') {
                        openingCount++;
                }
                if(exp[i] == ')') {
                        openingCount--;
                        if (openingCount < 0) {
                                return 0;
                        }
                }
                i++;
        }

        return openingCount == 0;
}
```

End of exam