THE UNIVERSITY OF BUCKINGHAM

SCHOOL OF SCIENCE

Exam Title: **PROBLEM SOLVING AND PROGRAMMING 1**

SPRING 2018

EXAMINATION FOR THE DEGREE PROGRAMMES IN

Computing, October 2017 entry and January 2018 Entry

Exam Code:	20182AIAP112AWP
TIME ALLOWED:	**2 HOURS**
MATERIALS PERMITTED:	None
MATERIALS PROVIDED:	ExamReferencePSP on P drive

INSTRUCTIONS:

1. Answer ALL questions.
2. Questions 1 to 3 require only the practical programming work whereas Question 4 requires both written work in the answer book and the practical programming work.
3. You must save all your practical work on the designated folder on the desktop of your computer.
4. You must organize your answers into projects: **one project** for **each question**.
5. **Do not** provide more than one answer to each question.
6. You may use the file “ExamReferencePSP” on the P drive for a quick reference of C/C++ language syntax.

Question 1 (15 Marks)

A hotel charges their customer's room rate for per night of stay and for the use of services such as breakfast and laundry. Valued Added Tax (VAT) of 20% is added to the total room cost and the services used.

Write a program that reads the room rate, the number of nights of stay and the total amount of services used and displays the total bill amount that a customer should pay. A sample run of the program should produce the following dialogue with the end user:

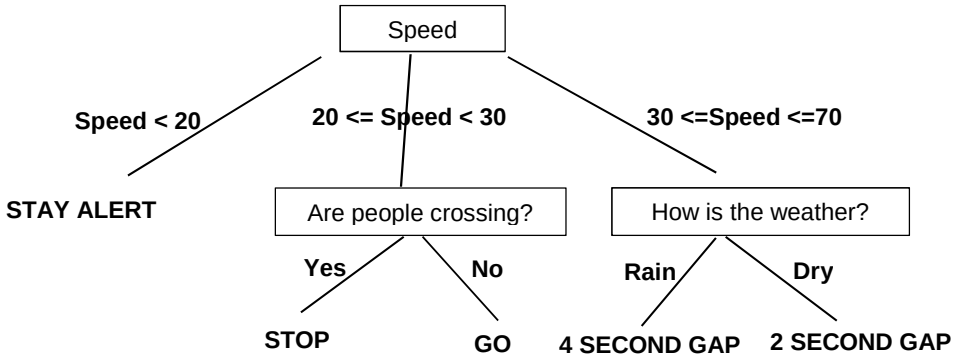
```
Enter the room rate: 75.5
Enter the number of nights of stay: 4
Enter the amount of services used: 135

Total amount for room and services: 437
VAT charged at 20%: 87.4
Total amount to pay: 524.4
```

Question 2 (15 Marks)

Driverless cars used to be the sort of thing you would see in science fiction films, but they are becoming a reality since 2017. In this exercise, we mimic very simple decisions of an autonomous car.

Write a program that receives necessary data inputs, applies the following decision scheme to the data inputs and classifies car actions as being either **STAY ALERT, STOP, GO, 4 SECOND GAP** or **2 SECOND GAP**.



Note that, the program can use characters to represent the input data, for example, character 'R' for Rain, 'D' for Dry, 'Y' for Yes, and 'N' for No. Full marks could be awarded if the program only asks necessary questions for specific groups.

For example: if the user enters the speed is 29. Then, the program will only ask "Are people crossing?" Then, if the user enters 'Y', the program will display "STOP".

Question 3 (20 Marks)

Write a local function to evaluate the below equation. This function receives one positive integer and returns a value. If the input of 'n' is 1, then the function returns value 1, and if the input is greater than or equal 2, then it returns a value calculated based on the below equation.

$$p = 1 + \frac{1}{2} + \frac{2}{3} + \frac{3}{4} + \cdots + \frac{n-1}{n}$$

The main program serves as a tester for the function developed; it will take input from users, check valid values, pass values onto a function call and display the result.

For example: The below table shows values of n and their correspondent values of p

n	p
1	1
2	1.5
3	2.16667
6	4.55

Question 4 (50 marks)

This question is concerned with developing and testing a library of functions (i.e. remote functions). It requires you to write your solution for each function in either **pseudo code or flowchart** in the answer book, construct the program components, and then test if the program components work.

Create a library called "mathLib"; this library should provide **three** functions, all of which work on Integer numbers. The purpose of each function is listed as following:

a) countDigits defines a function that counts the number of digits of a given positive integer. It receives one integer as an input and returns an integer as the total digits. **For instance**, enter an integer 618, the function will return 3.

Note that, full marks could be awarded if this function is a recursive one.

(15 marks)

b) evalPower defines a function that calculates the power of an integer number, i.e. $\mathbf{x^n}$. This function receives 2 integers as a base **x** and an exponent **n** and returns 1 integer.

Note that, full marks could be awarded if this function is a recursive one.

(15 marks)

c) isArmstrong function checks whether a positive integer is an Armstrong number. An Armstrong number is an **n-digit** number that is equal to the sum of the $\mathbf{n^{th}}$ power of its digits. This function receives one input and returns one boolean. ***true*** if the given number is an Armstrong number; otherwise ***false***.

For example: 6 has a **1**-digit number, and $6^{\mathbf{1}} = 6$

407 have **3** digits, and $4^{\mathbf{3}} + 0^{\mathbf{3}} + 7^{\mathbf{3}} = 407$

9474 have **4** digits, and $9^{\mathbf{4}} + 4^{\mathbf{4}} + 7^{\mathbf{4}} + 4^{\mathbf{4}} = 9474$

To obtain full marks, this function should show how to use the other two functions and is **NOT** necessary to be a recursive function.

(15 marks)

You should follow the normal steps of developing library functions, i.e. function declarations in **mathLib.h**, function definitions in **mathLib.cpp**, and **function calls** in the main program.

(5 marks)

--- END OF PAPER ---