



FINAL ASSESSMENT/EXAMINATION

SEMESTER 1 (2019/2020)

Course Code and Title: PROG1004 Computer Programming I

Programme: *B.A.Sc. in Computer Engineering.*

Date: 05/12/2019 **Time:** 1:00PM - 4:00PM

Duration: three (3) hours

PLEASE READ ALL INSTRUCTIONS CAREFULLY BEFORE YOU BEGIN THIS EXAMINATION

Instructions to Candidates

1. This paper has 6 pages and 6 questions for a total of 55 marks.
2. You are required to answer ALL questions.
3. Questions do not carry equal marks
4. You are allowed to use a non-programmable calculator
5. Complete the exam in pen.
6. Answer each question on a new page in your answer script
7. You are required to return the question paper with your answer script.

Key Examination Protocol

1. Students please note that academic dishonesty (or cheating) includes but is not limited to plagiarism, collusion, falsification, replication, taking unauthorised notes or devices into an examination, obtaining an unauthorised copy of the examination paper, communicating or trying to communicate with another candidate during the examination, and being a party to impersonation in relation to an examination.
2. The above mentioned and any other actions which compromise the integrity of the academic evaluation process will be fully investigated and addressed in accordance with UTT's academic regulations.
3. Please be reminded that speaking without the Invigilator's permission is **NOT** allowed.

| QUESTION | MARK |
|----------|------|
| 1 | / 5 |
| 2 | / 8 |
| 3 | / 10 |
| 4 | / 10 |
| 5 | / 12 |
| 6 | / 10 |
| <hr/> | |
| TOTAL | / 55 |

QUESTION 1 (5 marks)

Write a C program to determine the number of occurrences of the letters *v* and *q* in a sentence. The input consists of a sentence which is made up of alphabetic characters and spaces only and which terminates with a period. Note that the letters *v* and *q* may be in either uppercase or lowercase.

QUESTION 2 (8 marks)

What would be the output from the following program?

Note printf_s is the same function as printf

```
#include <stdio.h>

void main()
{
    int x = -5, y, z = 5;

    while (x <= 2)
    {
        y = 6;
        while (y <= 9)
        {
            z += 5;
            y += 2;
        }
        x += 2;
    }
    printf_s("z = %d \n", z);
}
```

QUESTION 3 (10 marks)

The Taylor expansion for determining $\ln(1 + x)$ is:

$$\ln(1 + x) = x - x^2/2 + x^3/3 - x^4/4 + \dots$$

where $\text{abs}(x) < 1$.

Your job is to write a C program to determine $\ln(1 + x)$ for a given value of x whose absolute value is less than 1. The input consists of the value x . Apply the Taylor expansion to 20 terms in order to determine / output the required value. Note that you do **not** have to validate that the absolute value of x is less than 1.

QUESTION 4 (10 MARKS)

This question requires you to develop a C program to determine the area bounded by the graph of a curve defined by

$$f(x) = 3x^2 - 4$$

and the x-axis.

For this question, use the trapezoid method to approximate the area.

QUESTION 5 (12 marks)

You have been asked to develop a C program for UTT's Volunteerism programme. This programme allows students to volunteer at charitable organizations and to have the hours worked at these organizations credited towards their degree.

Your job is to write a program to read in a student's id (integer). After entering the id for a student, your program should read in the id (integer) of a charitable organization at which the student worked followed by the number of hours worked at that organization. The student might have worked at several charitable organizations. Your program should therefore ask if there are any more inputs for this student. If the response is Y (for Yes), then your procedure should ask for the id of the next charitable organization followed by the number of hours worked at that organization. This process should continue until the response is N, indicating that there are no more charitable organizations at which this student worked. After processing all the inputs for a given student, your procedure should output the student's id along with the total number of hours worked at the various organizations.

After processing all the data for one student, your program should ask if there are any more students to be processed. Continue the above process as long as the response to this question is Y.

Your program should validate all responses to questions – Y for Yes or N for No. Also, the responses to the questions may be in either uppercase or lowercase.

QUESTION 6 (10 marks)

An instructor has manually maintained an *absenteeism* list of the students in his class. The list consists of student id's (integer), session numbers (integer), and whether they were present, absent or late for the session. The list has the following format:

| <u>Student Id</u> | <u>Session Number</u> | <u>P, A, or L</u> |
|-------------------|-----------------------|-------------------|
| 123 | 1 | P |
| 123 | 2 | L |
| 123 | 3 | P |
| . | | |
| . | | |
| . | | |
| 123 | 12 | L |
| 256 | 1 | P |
| 256 | 2 | P |
| 256 | 3 | L |
| . | | |
| . | | |
| 256 | 12 | P |
| . | | |
| . | | |
| . | | |

As you can see, for each student there are twelve entries, one for each session indicating whether the student was present (P), absent (A) or late (L). Assume that there are twenty students in the class.

Your job is to develop a C program to computerize this list for the instructor using three appropriately defined one-dimensional arrays. Having defined the arrays, write the code to populate the arrays. Finally, your program must be able to enter a student id then determine and output the number of times this student was absent during the semester.

END OF EXAMINATION