Department of Computer Science

COS284 Practical and Assignment 4: Functions

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

1 Introduction

This document contains both practical 4 and assignment 4. In general the assignments will build upon the work of the current practical.

1.1 Submission

The Practical will be fitchfork marked but the tutors will be available for assistance during the practical sessions on 3 and 4 September. Upload your practical before submissions close at 23:59 on the 4th of September.

The Assignment will be fitchfork only and is due at 17:00 on Friday 14th of September. You may use the practical sessions to work on your assignment in the labs and ask for assistance if it is required.

1.2 Plagiarism policy

It is in your own interest that you, at all times, act responsible and ethically. As with any work done for the purpose of your university degree, remember that the University of Pretoria will not tolerate plagiarism. Do not copy a friend's work or allow a friend to copy yours. Doing so constitutes plagiarism, and apart from not gaining the experience intended, you may face disciplinary action as a result.

For more on the University of Pretoria's plagiarism policy, you may visit the following webpage: `http://www.library.up.ac.za/plagiarism/index.htm`

1.3 Practical component [25%]

There are 3 tasks for this practical and it will be fitchfork marked only.

1.3.1 Task 1: Addition and Odd Checker [5]

For this task you will be given a file driver.c as well as a makefile, you must implement an assembler program that does the following:

- Implement a function named addToFindEvenOrOdd in assembly which must receive two signed integer parameters.
- The function must then perform addition on the two parameters.
- The function must further return a one(1) if the result is an odd number, and a zero(0) otherwise.

For example, the function might receive 3 and 4, which it must add together, getting the result 7. Since 7 is an odd number, the function must return 1.

When you are finished, upload your source code file named **task1.asm** to the assignments.cs.up.ac.za website, under the **Practical 4 Task 1** submission slot.

1.3.2 Task 2: Dot Product [10]

For this task you must implement an assembler function that computes the dot product of two 2d vectors. You should:

- Create a function in assembly that computes the dot product, called dotProduct.
- It should accept 4 int parameters, one for each vector element.
- It should return a single number, the dot product.

You can find more information on the dot product at `https://www.mathsisfun.com/algebra/vectors-dot-product.html`

When you are finished, upload your source code file named **task2.asm** to the assignments.cs.up.ac.za website, under the **Practical 4 Task 2** submission slot.

1.3.3 Task 3: Palindrome [10]

For this task you have to implement a palindrome checker. The function will have to take in 6 characters in the parameters and will have to check if the word is the same if it were written backwards.

- The function has to be called palindromeCheck and will have to return an integer value of either 0 (false) or 1 (true).

- ABCCBA is a palindrome whilst CHEESE is not. You can assume that only upper-case letters will be used.
- Note: You will have to make a C file to test your function as well as a corresponding makefile.

For more information on palindromes follow this link:
`https://en.wikipedia.org/wiki/Palindrome`

When you are finished, upload your source code file named **task3.asm** to the assignments.cs.up.ac.za website, under the **Practical 4 Task 3** submission slot.

1.4 Assignment component [75%]

1.4.1 Task 2: Matrix Determinant [75]

For this assignment, it will be required of you to implement a function that takes 16 parameters and would then return the determinant of all of the parameters. You can see it as a 4 x 4 matrix that you would use(thus 16 parameters) to calculate this function. Thus you should note that you will have to use the stack in order to pass more than 6 parameters to a function.

- The function has to be called matrixDeterminant and will have to return an integer value.
- The parameters passed will be used in the following way: the first set of four will be the top row, the second set of four the second row, the third set of four will be the third row and thus the last set of four will be the final row.
- Note: You will have to make a C file to test your function as well as a corresponding makefile.

For more information on how to calculate the determinant of a matrix follow this link:
`https://en.wikipedia.org/wiki/Determinant`

When you are finished, upload your source code file named **assignment4.asm** to the assignments.cs.up.ac.za website, under the **Assignment 4** submission slot.

2 Mark Distribution

Activity	**Mark**
Prac Task 1	5
Prac Task 2	10
Prac Task 3	10
Assignment	75
Total	**100**