Computer Science 3B

Practical Assignment 04

2017-08-30

Deadline - 2017-08-30 17h00

Marks: 60

This practical assignment must be uploaded to eve.uj.ac.za **before** 2017-08-31 17h00. Late or incorrect submissions **will not be accepted**, and will therefore not be marked. You are **not allowed to collaborate** with any other student.

Good coding practices include a proper coding convention and a good use of commenting. Marks will be deducted if these are not present. See the reminder page for more details.

Write a complete x86 assembly program that creates a function for each of the following:

- A function that returns the sum of two parameters (sum-function).
- A function that asks the user to input n-items and return an array of n-items (array-function).
- A function that that outputs each of the array items, but adds a user-defined value (output-function)

Your main function should ask the user for two items:

- The number of items on the array.
- The value that will be added to each item of the array when the last function is used.

You program should have the following flow:

- Prompt the user for the size of the array and the offset address.
- Your main function should then call the array-function and prompt the user for each of the items in the array.
- Your main function should then call the output-function that makes use of the sum-function.

The only global variables allowed are string constants. Any other variables must be local variables.

You are not allowed to make use of the INVOKE command, make use of the CALL command for any input, output and calling of function.

Your design document should include C++ or java-code for each function, AS WELL as a stack frame for each function.

Mark sheet

1. Design (C++ or Java code)	[05]
2. Design (Stack diagrams)	[05]
3. Add-function (entry - 2, body - 1, exit - 2)	[05]
4. Array-function (entry – 2, body – 6, exit – 2)	[10]
5. Display-function (entry – 2, body – 1, exit – 2)	[05]
6. Main-function (user input – 2, local variables – 2, exit – 1)	[05]
7. Structure and layout (no extra globals, correct data types, no use of Invoke)	[05]
8. Commenting	[05]
9. Correct execution.	[15]

NB

Submissions which **do not assemble** will be capped at 40%!

Execution marks are awarded for a correctly functioning application and not for having some related code.

Reminder

Your submission must follow the naming convention as set out in the general learning guide.

Example

Surname	Berners-Lee	**Initials**	TJ
Student number	209912345	**Module Code**	CSC3B10
Current Year	2017	**Practical number**	P05

```
Berners-Lee_TJ_209912345_CSC3B10_2017_P05
```

Your submission must include the following in a single zip (compressed) file:

- *Source file* (`asm` file) - File containing your solution. Your details must be included at the top of the source code.
- *Program design* (`pdf` file) - File containing your design. Your details must be included at the top of the design.