Computer Science 3B

Practical Assignment 05

Assignment date:	2024-08-23
Deadline	2024-08-23 17h05
Marks: 150	

This practical assignment must be uploaded to eve.uj.ac.za **before** 2024-08-23 17h05. Late[1] or incorrect submissions **will not be accepted**, and will therefore not be marked. You are **not allowed to collaborate** with any other student. Plagiarism is not tolerated. All submissions are tested for plagiarism.

Good coding practices include a proper coding convention and a good use of commenting. Marks will be deducted if these are not present.

The reminder page includes details for submission. Please ensure that **ALL** submissions follow the guidelines. The reminder page can be found on the last page of this assignment.

BeatWatch wants to try to use 2D arrays but have decided to leave that for a later date. Instead **BeatWatch** has tasked you to create a more modular program to handle the different algorithms involved in their processes. The next formula is a ***Health Score***:

$$\textbf{HealthScore} = \frac{\text{level} \times \text{duration} \times \textbf{HRF}}{\frac{\text{age}}{10} + 5}$$

Write an **80x86** assembly program with the following:

Functions:

```
//  Fill an array with user input
void input(int* array, int length);
//  Display the contents of an array
void display(int* array, int length);
//  Calculate health score as per formula
int healthScore(int age, int level, int duration, int hrf);
//  Calculate scores from inputs
void finalScore(int age, int* levels, int* durations, int hrf, int* scores);
```

Main program structure:

1. Ask the user **age** and **HRF** using normal integer input
2. Ask the user for **level**[2] and **duration**[2] arrays using the ***input*** function
3. Display the **age** and **HRF** using normal integer output
4. Display the **level** and **duration** arrays using the ***display*** function.
5. Calculate the **score**[2] array in the ***finalScore*** function, making use of the ***healthScore*** function
6. Display the **score** using the ***display*** function
7. Ask the user if they want to process another set of values. If they do go back to step 1.

[1] Alternate arrangements for exceptional circumstances will been posted on eve.

[2] All arrays are of length 5

You are only allowed to create global variables the **level**, **duration** and **score** arrays as well as any string prompts. All other user inputs must be stored in local variables. Do not perform any rounding.
Note: The design must include a stack diagram for each function as well as **_start**.

Testing set - Use these values to test your program:

Age	HRF	Level	Duration	Score
18	2	[1,3,5,1,4]	[144, 20, 54, 53,164]	[48, 20, 90, 17, 218]
23	58	[1,2,2,5,2]	[23,133,103,152,138]	[190,2204,1706,6297,2286]
36	8	[1,5,3,3,1]	[85,173, 47, 99,122]	[85, 865, 141, 297, 122]
47	46	[1,3,1,5,4]	[170, 45, 26,164,107]	[868, 690, 132,4191,2187]
59	36	[4,1,5,2,1]	[50,166,102,163,122]	[720, 597,1836,1173, 439]

Bonus

Submit the bonus project as a separate zipfile with the practical number as P05_B. Failure to do so will result in issues with marking the normal practical and lead to a loss of marks!

Ask the user for the size of the userArray and create the array dynamically on the stack.

Mark sheet

1.	Design	[10]
2.	***input*** function (entry/exit, input, loop)	[15]
3.	***display*** function (entry/exit, output, loop)	[15]
4.	***healthScore*** function (entry/exit, calculation, loop)	[15]
5.	***finalScore*** function (entry/exit, calculation, loop)	[20]
6.	Main procedure	[15]
7.	Structure and layout (no temporary variables, correct data types)	[05]
8.	Commenting	[05]
9.	Correct execution	[50]
10.	Dynamic Arrays	[20 (bonus)]
	Total	[150]

NB

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

Practical marks are awarded subject to the student's ability to explain the concepts and decisions made in preparing the practical assignment solution.

(Inability to explain code → inability to be given marks.)

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

Reminder

Your submission must follow the naming convention below:

`SURNAME_INITIALS_STUDENTNUMBER_SUBJECTCODE_YEAR_PRACTICALNUMBER`

Example: `Berners-Lee_TJ_209912345_CSC03B3_2024_P05`

Surname	Berners-Lee	**Module Code**	CSC03B3
Initials	TJ	**Current Year**	2024
Student number	209912345	**Practical number**	P05

Your submission must be **a single zip (compressed) file**!

Your submission must include the following:

File	**Naming**	**Folder**	**Purpose**
`Design`	`STUDENTNUMBER_P05.pdf`	`docs`	Contains your program design. All files must be in **PDF** format. Your details must be included at the top of any **PDF** files submitted[0].
`Source`	`STUDENTNUMBER_P05.asm`	`src`	Contains all relevant source code. Your details must be included at the top of the source code[0].

Multiple uploads

Note that only **one** submission is marked. If you already have submitted once and want to upload a newer version then submit a newer file with the same name as the uploaded file in order to overwrite it.

[0] Failure to correctly indicate your details will result in a penalty.