NAMIBIA UNIVERSITY
OF SCIENCE AND TECHNOLOGY

Faculty of Computing and Informatics

Computer Science Department

DSA521S Project

This assignment consists of 2 parts, Section A and Section B. Section A is for pseudocode representation and Section B is for the practical implementation of a working solution. Each group MUST consist of between 7 and 10 members. All project documents and solutions MUST done on GitHub and links to each group's groups repository shared on eLearning. Lastly, kindly use the marking rubric given in each section to determine the amount of effort required on each component.

SECTION A: Algorithms representation of the different modules and/functions (Pseudocode/flowchart) [75 marks]

Project summary

A Namibian music start-up is looking for an efficient music player algorithm for a mobile application. Suppose you have 100 tracks on your list. The playlist works on the concept of linked list. Tracks are played one by one (best example of singly linked list. Track are connected and you can move from track four to five but you cannot go back.

1. Implement functionality to play tracks in both directions (following the behaviour of double linked list).
2. In addition, add the functionality to allow for the playing of tracks on repeat (circular).
3. Implement the functionality to add and remove tracks from playlist
4. The application should also allow for searches on playlist. E.g a user could search for a specific track

Using ideas from class, design an algorithm to implement the functionalities described in the project summary. As far as possible, design and specify different modules/functions that make up the application.

Kindly use the marking rubric below to design software that can assist the music start-up. (Note: Students are allowed to expand the scenario as long as it remains in the confinement of music start-up operations). The rubric is only a guide and should not constrain you from any other design ideas you may have.

Criteria	Poor Work	Below Average	Average Work	Above Average	Exceptional Work
*Modules **(9)***	*No modules exists at all **(0)***	*Modules exists but do not help much in solving the problem **(3)***	*At least three suitable modules are clearly Distinguished. **(5)***	*More than three clearly spelt modules exists **(7)***	*Brilliant decomposition applied makes the problem very easy to solve. **(9)***
*Functions **(15)***	*Functions poorly defined and do not help much in solving the problem **(3)***	*Functions are defined but at times combines several purposes. **(6)***	*At least 5 well defined functions exist **(9)***	*More than 5 well-defined functions exist. **(13)***	*All functions defined are necessary and stick to their purpose. **(15)***
*Pseudocode **(25)***	*The pseudocode is not sound at all and it is difficult to follow. **(5)***	*Although sound, the pseudocode is lengthy and not precise. **(10)***	*The logic of the pseudocode is sound and the logic easy to follow. **(15)***	*Sound and precise pseudocode with some comments on it. **(20)***	*Pseudocode presented in modules, also sound, precise, and easy to follow. **(25)***
*Flowchart **(16)***	*A flowchart presented is flawed with wrong symbols and no software used. **(3)***	*Although appropriate software was used, some symbols were wrong. **(6)***	*A sound flowchart with all components exists using the right symbols. **(9)***	*Appropriate software used to develop a sound flowchart with right symbols. **(12)***	*Connectors used to join a sound flowchart using suitable software. **(16)***
*Documentation **(10)***	*No README file in the repository. **(0)***	*A README document exists in the repository but does not clearly describe the project. **(3)***	*A well-documented README exists in the git repository clearly explaining what the project does. **(5)***	*A README document briefly describes the solution in general and also the modules available. **(7)***	*Apart from general description, modules and functions are also described with names of contributors available. **(10)***
TOTAL	***11***	***28***	***43***	***59***	***75***

NB: The marks in **bold** indicate the maximum that a student can attain under each category, meaning a lower mark can be attained to the evaluator's discretion. This applies for both Section A and B.

NOTES:

SUBMISSION DEADLINE: Group Assignment - 18 October 2022 @ 23h59
ONLY 1 Group member to submit on behalf of all members with students numbers and names of all Member clearly stated on the project.

SUBMISSION CRITERION: Links to solutions for both Section A and B are to be shared on e-learning for assessment purposes from GitHub.

SECTION B: Practical implementation of the program designed in section A. [25 Marks]

Use ideas from programming to implement your solution algorithm designed in section A. Students are free to use any programming language of their choice to implement the data structure and the operations specified.

Criteria	**Below Expectation**	**Met Expectation**	**Surpass Expectation**
Creativity (7)	*Dull project that do not fit a first year student at university. The project seems likes blocks were just thrown without making a good meaningful project.* ***(2)***	*The project is meaningful and the flows are acceptable at NQF5 even though some improvements could have been made.* ***(5)***	*Choice of colours, sounds, and other components fuse seamless to make one fantastic project above NQF5* ***(7)***
Logic (11)	*Simple sequential control structures are basically used in the project and nothing more.* ***(4)***	*A number of logical operations exists in the project and the whole project is sound.* ***(8)***	*Some interesting programming constructs exists in the project.* ***(11)***
Effort (7)	*The project seems like it is a weekend's effort without much time dedicated to the project.* ***(2)***	*A reasonable amount of time satisfactory to the evaluators seem to have been committed to the project.* ***(5)***	*The amount of time put in the project surpasses the evaluators expectation.* ***(7)***
TOTAL	***8***	***18***	***25***