

***DATA STRUCTURE & ALGORITMS GROUP ASSIGNMENT***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Group no** | **Name** | **Student Number** | **Specialization** | **Programme** |
| 7 | Lerise.M.Skrywer (**group leader)** | 222085630 | Informatics | Full- time |
| 6 | Carlton Hishiko | 222129514 | Computer Science | Full - time |
| 1 | Emilia Haufiku | 222093439 | Cyber Security | Part- time |
| 1 | Elizabetha Taapopi | 220127824 | Cyber Security | Full - time |
| 1 | Mika Haindongo | 222133449 | Computer Science | Part - time |

**SECTION A**

**1.**

initialize music player

initialize playlist

initialize tracklist

initialize repeat function

initialize search function

initialize add/remove function

play music player

play tracks from playlist

repeat tracks as necessary

search for specific tracks

add/remove tracks as necessary

exit

end

**2.**

function create\_playlist()

tracks := 100

linked\_list := create\_new\_linked\_list()

for i := 0 to tracks-1 do

add\_track(linked\_list, i)

end for

return linked\_list

end function

function play\_track(linked\_list, track\_number)

current\_track := linked\_list.head

for i := 0 to track\_number-1 do

current\_Track := current\_Track.Next

end for

play\_audio(current\_track.data)

end function

3.

function add\_track(linked\_list, track\_number)

new\_track := create\_new\_track(track\_number)

if linked\_list.head = null then

linked\_list.head := new\_track

else

current\_track := linked\_list.head

while current\_track.next != null do

current\_track := current\_track.next

end while

current\_track.next := new\_track

end if

end function

function remove\_track(linked\_list, track\_number)

current\_track := linked\_list.head

previous\_track := null

for i := 0 to track\_number-1 do

previous\_track := current\_track

current\_track := current\_track.next

end for

if previous\_track = null then

linked\_list.head := current\_track.next

else

previous\_track.next := current\_track.next

end if

end function

4.

function search\_playlist(linked\_list, track\_number)

current\_track := linked\_list.head

while current\_track != null do

if current\_track.data = track\_number then

return true

end if

current\_track := current\_track.next

end while

return false

end function

**SECTION B**

function create\_playlist()

tracks := 100

linked\_list := create\_new\_linked\_list()

for i := 0 to tracks-1 do

add\_track(linked\_list, i)

end for

return linked\_list

end function

function play\_track(linked\_list, track\_number)

current\_track := linked\_list.head

for i := 0 to track\_number-1 do

current\_track := current\_track.next

end for

play\_audio(current\_track.data)

end function

function add\_track(linked\_list, track\_number)

new\_track := create\_new\_track(track\_number)

if linked\_list.head = null then

linked\_list.head := new\_track

else

current\_track := linked\_list.head

while current\_track.next != null do

current\_track := current\_track.next

end while

current\_track.next := new\_track

end if

end function

function remove\_track(linked\_list, track\_number)

current\_track := linked\_list.head

previous\_track := null

for i := 0 to track\_number-1 do

previous\_track := current\_track

current\_track := current\_track.next

end for

if previous\_track = null then

linked\_list.head := current\_track.next

else

previous\_track.next := current\_track.next

end if

end function

function search\_playlist(linked\_list, track\_number)

current\_track := linked\_list.head

while current\_track != null do

if current\_track.data = track\_number then

return true

end if

current\_track := current\_track.next

end while

return false

end function

public class musicPlayer {

public static void main(String[] args) {

}

public final class DoubleClass<Music> {

private Music data;

private DoubleClass<Music> next;

private DoubleClass<Music> prev;

public DoubleClass(final DoubleClass<Music> prev, final Music data, final DoubleClass<Music> next) {

this.data = data;

this.next = next;

this.prev = prev;

}

public DoubleClass(final Music data) {

this(null, data, null);

}

public Music getData() {

return data;

}

public DoubleClass<Music> getNext() {

return next;

}

public void setTheNextTrack(final DoubleClass<Music> next) {

this.next = next;

}

public void setThePreviousTrack(final DoubleClass<Music> prev) {

this.prev = prev;

}

public DoubleClass<Music> getPrev() {

return prev;

}

**public void addTracks(final Music data) {**

**this.data = data;**

**}**

**}**

**}**

|  |  |  |  |
| --- | --- | --- | --- |
| **Group no** | **Name** | **Student Number** | **TASK** |
| 7 | Lerise.M.Skrywer (**group leader)** | 222085630 | Did section B and compiled everything |
| 6 | Carlton Hishiko | 222129514 | Did section A 3 |
| 1 | Emilia Haufiku | 222093439 | Did section A 2 and also helped with coding for section B |
| 1 | Elizabetha Taapopi | 220127824 | Did Section A 4 |
| 1 | Mika Haindongo | 222133449 | Did section A 1 |