**Question 1**

**Task 1.1**

● get all records from file

● store records in suitable data structure

● keep track of reasons and count of invalid passwords

● loop through each character

● … length check min 8 characters

● … check lowercase

● … check uppercase

● … check digit

● … check symbol

● … proper messages for each case

● display usernames, password, reasons …

**Question 2**

**Task 2.1**

● terminating case: n=1 and n=2

● recursive call fib(n-1) + fib(n-2)

● parameter and return value

● loop to print first 15 Fibonacci numbers

● screenshot

**Task 2.2**

● base case: n=1 and n=2

● initialise first two numbers to 1

● loop to add to total

● loop to print first 15 Fibonacci numbers

● screenshot

**Task 2.3**

● method to count number of times and display number

● Screenshots for n=10 and n=30

● Recursive function -> use of stack

● Pushing & popping of variables, return values

**Question 3**

**Task 3.1**

● Class Node

● … Word, Count, Pointer

● … get and set methods

● constructor

● … attributes Start, NextFree

● … Wordlist array of 20 nodes with Pointers pointing to next node

● Display header and content and pointer in index form

**Task 3.2**

● main program with instance of LinkedList and call Display

**Task 3.3**

● check if new\_value exist…

● …exist: find node and increment count

● …non-existent: check for free nodes, error msg

● Grab new node from free list set Word to new\_value and Count to 1

● Case: insert into empty list and change pointers

● Search for position to insert new node using pointers and update pointers

● Table of additional method/variable with explanation

**Task 3.4**

● get all records from file: open, read, close, into suitable data structure

● call Insert to insert each word

● Display linked list

**Task 3.5**

● ReverseTraversal with parameter

● … check for NULL

● … call recursive function

● … print words and count in reverse order

● Call function in main program with Start index

**Question 4**

**Task 4.1**

● Class SongRecord, songID and songTitle

● … get and set methods

**Task 4.2**

● Class HashTable: constructor

● Size=13

● Array with 13 slots

● Hash method with parameter SongID

● …Calculate total ASCII value and return Remainder

● Display method with heading

● Add method with parameters

● …get address with Hash function and add record

● Remove method with parameter

● …get address with Hash function and remove record

**Task 4.3**

● screenshot of adding and removing records

**Task 4.4**

● loop to check for empty slot and existance of SongID

● … appropriate message for same SongID

● loop to find SongID…

● …and complete one round of array size

● … appropriate message for same SongID

● Linear probing: collision of records

● Search sequentially until next available

**Task 4.5**

● screenshot of adding and removing records