# Eksamen i PG3401 C Programmering

# Generelt om koden:

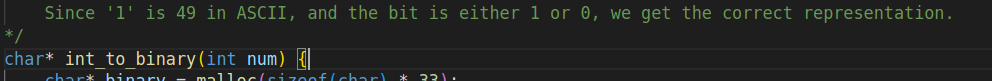
- Koden er skrevet på engelsk

- Casingen jeg har brukt er snake\_case som er en «uskrevet» standard for C

- Noe av koden som er nedlastet fra eksamens teksten er modifisert for å passe denne standarden

Et lite sidenotat som jeg følte jeg måtte nevne:

Jeg har prøvd å unngå å bruke «We» i kommentarer ettersom oppgaven er individuell, men ser jeg har brukt det noen steder der jeg føler det er naturlig:



# Oppgave 1

## Forklar hva C programmeringsspråket kan brukes til

C er et såkalt «lowlevel» programmeringsspråk og kan brukes til så å si alt andre programmeringsspråk kan. Det er også fordeler og ulemper med om språket er lowlevel eller highlevel, lavnivå eller høynivå. Språk som er av typen lavnivå kan sies at er nærmere maskinkode og maskinvare.

Grunnet at språket er av typen lavnivå er det mange bruksområder. C kan brukes til å skrive operativsystemer (OS), databaser (DB), databasehåndteringssystemer (DBMS), drivere og innebygd system (embedded system).

## Hvem er Dennis Ritchie og hva er han kjent for innen Informasjonsteknologi?

Dennis Ritchie var en amerikansk data programmerer/informatiker. Først og fremst er Dennis kjent for å skape programmeringsspråket C og være en av medskaperene til Unix. Dennis jobbet hos Bell Labs utviklet han sammen med kolleger operativsystemet Unix, på slutten av 60-tallet. Unix var originalt skrevet i Assembly, men ble etter kort tid omskrevet til C.

## Forklar hva kommandoen sudo gjør i terminal på Linux, og hva man typisk bruker denne kommandoen til ved bruk / administrasjon av Linux

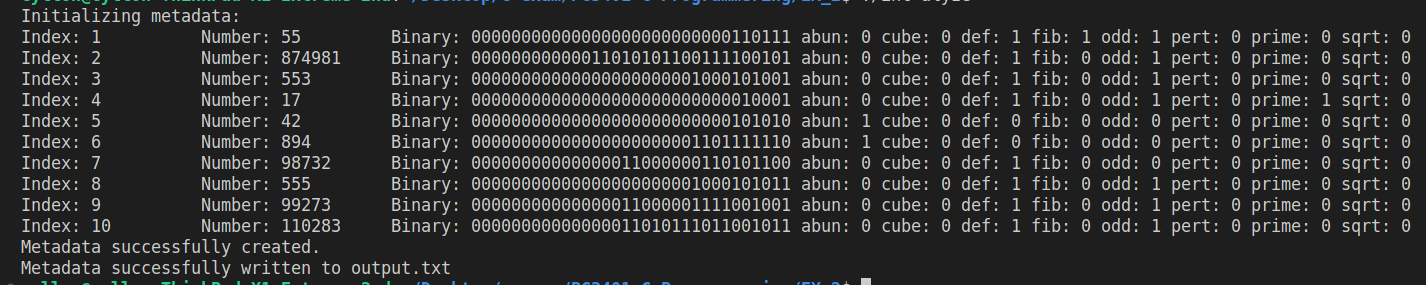
Kommandoen «sudo» er en forkortelse for «superuser do» og ved å bruke denne kommandoen blir rettighetene dine midlertidig hevet til administratorrettigheter. Er som oftest bruk for å gi rettigheter til filer i et linux-basert system.

# Oppgave 2

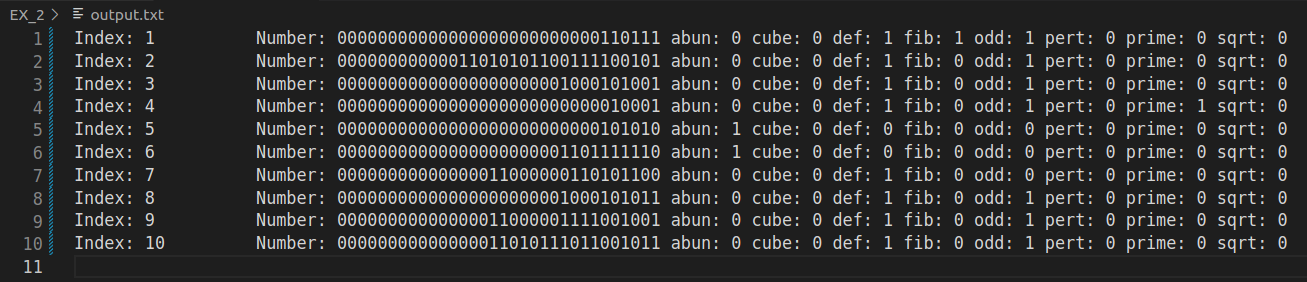
Eneste jeg var usikker på her var om hele output’en skulle konverteres til binært eller bare nummeret. Slik jeg tolket det var det bare nummeret.

Den kjørbare filen heter: «int-alyze».

Programmet kan kjøres slikt: «./int-alyze»

Output i terminal ser slik ut:

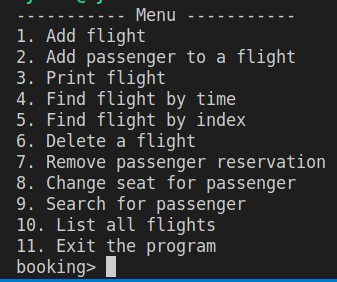
Output i fil ser slik ut:



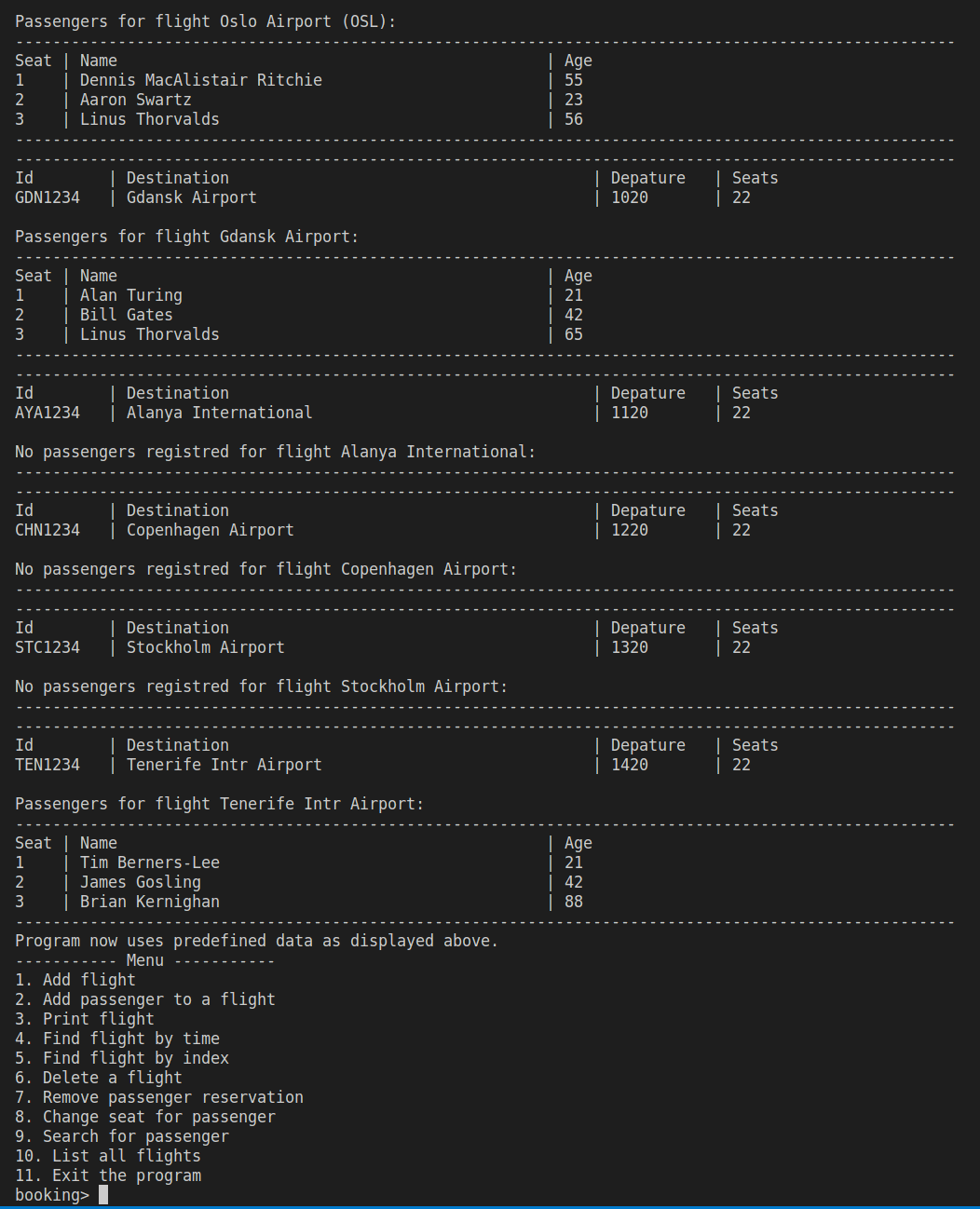
# Oppgave 3

Den kjørbare filen heter: «flbook».

Programet kan kjøres slik uten predefinert data: «./flbook». Da ser programmet slik ut ved oppstart:



Programmet kan kjøres slik med predefinert data: «./flbook -predef». Da ser programmet slik ut ved oppstart:



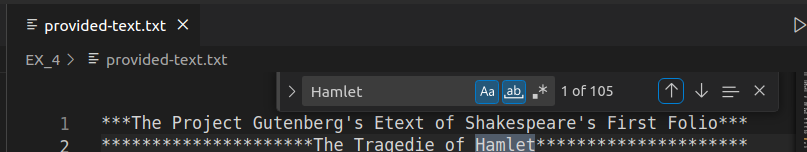
# Oppgave 4

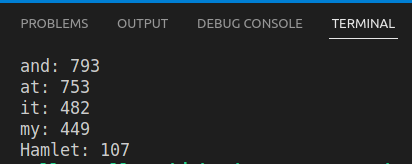
Den kjørbare filen heter «text-alyze».

Programmet kan kjøres slikt: «./text-alyze provided-text.txt»

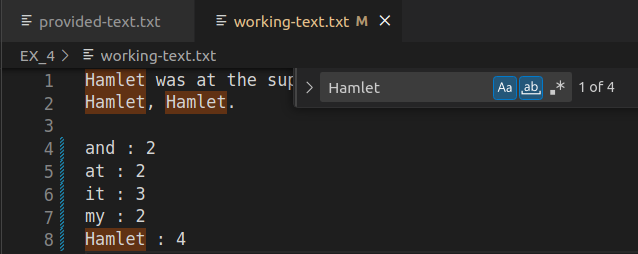
Telling av ASCII tegn fra 32 til 126 fungerer som tenkt.

Når jeg kjører oppgave 4 med teksten gitt fra lenke i eksamentekst(provided-text.txt) virker det som at tellingen av forekomster av ord ikke er likt som hva min editor sier når jeg søker etter samme ordet:

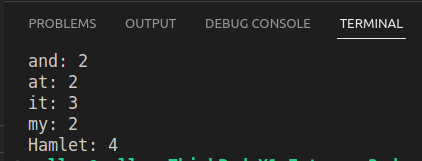
Output fra program:



Men når jeg kjører en tekst jeg komponerte selv(working-text.txt), så later det til å fungere:



Output fra program:



Dette problemet kommer enten av hvordan jeg sammenligner ordene, kode-editoren som teller feil, tegn-sett eller lignende.

# Oppgave 5

Den kjørbare filen heter: «rev-shell».

Programmet kan kjøres slikt for server:

«sudo ./rev-shell -listen -port <port>».

Programmet kan kjøres slikt for klient:

«sudo ./rev-shell -server <host> -port <port>».

Hva som ikke fungerer:

- Når man skrur av klienten med <CTRL + C> så skrur ikke serveren seg av. Prøvde å lage en slags ping mekanisme.

# Oppgave 6

Den kjørebare filen heter «codebf».

Programmet kan kjøres slikt: «./codebf <fil\_navn.c>»

Oppgave 6 ble litt i det vanskeligste laget. Fikk bare til underoppgave C).