

Oblig 2 dat103

Oppgave 1:

T	0	1	2	3	4	5	6	7
S	0	1	1	1	0	1	1	0
R	1	0	1	0	1	1	0	0
Q	1	0	0	0	1	1	0	X (invalid stat)

Oppgave 2.1:

- (a) Immediate (Umiddelbar): Immediate trenger 0 hukommelsetilgang.
- (b) Direct (Direkte): Direct trenger 1 hukommelsetilgange.
- (c) PC relative (PC relativ): PC relative trenger 2 hukommelsetilganger.
- (d) Register: Register krever 1 hukommelsetilgang.

Oppgave 2.2:

- A) Immediate: har ingen effektive adresse og har operand 200
- B) Direct: har effektive adressen 200 og har operand 400
- C) Base-Register: har den effektive adresse $99+200=299$ og har operand 899
- D) PC-Relative: har effektive adressen $200+101=301$ og har operand 901
- E) Indirect: Indirect går til adressen 200 som har effektiv adresse som 400 og har operand 1000.
- F) Register RR: har effektive adresse RR og har operand 199
- G) Register indirect RR: har adresse RR og effektiv adresse 199 og har operand 1000

Gdb: debugger.

Oppgave 3.2

Utklipp for siffer mindre enn 10:

```
sindre@sindre-VirtualBox:~/oblig2/assembly$ ./summer
Skriv to ensifrede tall skilt med mellomrom.
Summen av tallene maa vaere mindre enn 10.
3 6
9
```

Utklipp for siffer max 18:

```
sindre@sindre-VirtualBox:~/oblig2/assembly$
sindre@sindre-VirtualBox:~/oblig2/assembly$ ./summer2
Skriv to ensifrede tall skilt med mellomrom.
Summen av tallene maa vaere mindre enn 18.
9 9
18

sindre@sindre-VirtualBox:~/oblig2/assembly$
sindre@sindre-VirtualBox:~/oblig2/assembly$
```

Oppgave 3.3

```
sindre@sindre-VirtualBox:~/oblig2/assembly$ ./javaloop
@sindre@sindre-VirtualBox:~/oblig2/assembly$
```