

### Oppgave 1:

a)

For variabel i og j stemte det godt. Fikk feil på tegningen for \*p og \*q, men etter å ha sett utskriftene ser jeg hvor jeg tenkte feil. Det blir naturligvis «omvendt» når man gir en variabel med \* en «vanlig» verdi som for eksempel \*p = &i => \*p = 3. Da blir det også naturlig at &p vil lede til adressen for den nye variabelen.

b)

Her ble resultatet p = 8 & q = 8, som gir mening etter litt resonering. Mine tegninger kom frem til at begge fikk verdi 7 etter linjen p = q, men det ble 8. Har lagt kommentarer inn i koden der jeg følger med på endringen etter hver linje.

### Oppgave 2:

```
char *line = nullptr;    // eller char *line = 0;
strcpy(line, "Dette er en tekst");
```

Den gitte koden vil returnere en Core Dump feil som er en spesifikk type feilkode. Det er forårsaket av segmenteringsfeil, da man prøver og aksessere minne som koden ikke har tilgang til. Det kan ofte skje av at minneadressen er i ett «read-only» segment.

Feilkoden kan også oppstå ved korrupt minne, men i dette tilfellet skjer det på grunn av at metoden strcpy prøver å kopiere en udefinert stringverdi og den definerte teksten. Det vil naturligvis gi feilkode.

### Oppgave 3:

```
char text[5];
char *pointer = text;
char search_for = 'e';
cin >> text;
while (*pointer != search_for) {
    *pointer = search_for;
    pointer++;
}
```

Først og fremst kan while-løkken ende opp i en endless loop, da det ikke er definert noe slutt. Det vil også oppstå problemer ved å utnytte både \*pointer og pointer, da de ikke har samme verdi, og ikke vil gi forventet resultat.

### Oppgave 4:

Se kodefil med navn «Oppg4.cpp», har skrevet kommentarer som forklarer nok for min del hvertfall.

### Oppgave 5 & 6:

Se kode.