



**2b** (vekt 10%)

Hvis de  $p$  prosessorene er koblet sammen som en ring, vennligst forklar, steg for steg, hvordan en all-to-all broadcast kan realiseres ved hjelp av en-til-en kommunikasjoner.

**2c** (vekt 5%)

Anta at følgende modell for tidsbruk gjelder for å sende en melding med  $m$  ord fra en prosessor til en annen

$$t_s + t_w m,$$

hvor  $t_s$  og  $t_w$  er to konstanter. Utledd tidsbruk-modellen for utførelse av en all-to-all broadcast på en ring av  $p$  prosessorer. Hver prosessor initielt har  $m$  ord som sin data.

**Oppgave 3** (vekt 10%)

Hva vil bli skrevet til skjermen etter kjøring av følgende OpenMP program med fire tråder? Forklar hvorfor.

```
int main (int argc, char *argv[])
{
    int    i, n;
    float a[100], sum;

    n = 100;
    for (i=0; i < n; i++)
        a[i] = i * 1.0;
    sum = 0.0;

#pragma omp parallel for default(shared) reduction(+:sum)
    for (i=0; i < n; i++)
        sum = sum + a[i];

    printf("    Sum = %f\n", sum);
    return 0;
}
```

**Oppgave 4** (vekt 25%)

Følgende definisjon gjelder matrise-vektor produkt  $y = A * x$ , hvor  $A$  er en  $n \times n$  matrise,  $x$  og  $y$  begge er en vektor av dimensjon  $n \times 1$ .

$$y_i = a_{i,1}x_1 + a_{i,2}x_2 + \dots + a_{i,n}x_n, \quad i = 1, 2, \dots, n.$$

(Fortsettes på side 3.)

**4a** (vekt 5%)

Skriv en seriell C funksjon

```
void mat_vec(int n, double** A, double* x, double* y)
```

som implementerer  $y = A * x$ .**4b** (vekt 5%)

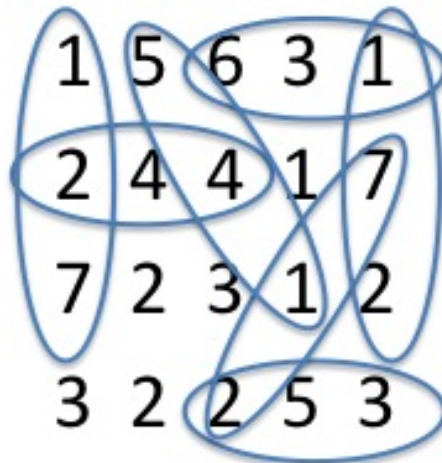
Paralleliser den serielle C funksjonen ved hjelp av OpenMP.

**4c** (vekt 15%)

Skisser en MPI implementasjon (vis bare de viktige detaljene), hvor initielt bare prosess 0 har verdiene til matrisen  $A$  og vektoren  $x$ . Sluttresultatet skal være at hele vektoren  $y$  er tilgjengelig på prosess 0.

**Oppgave 5** (vekt 25%)

Gitt en 2D tabell  $v$  som inneholder  $M \times N$  positive heltall, målet med denne oppgaven er å telle antall så-kalte “10-venner-tripler” i tabellen. En “10-venner-trippel” defineres som tre sammenhengende tall (horizontalt, vertikalt, eller på skrått) som summeres til 10. Noen eksempler av “10-venner-tripler” kan sees i figuren nedefor, markert med sirkler.

**5a** (vekt 10%)

Skriv en seriell C funksjon

```
int count_friends_of_ten (int M, int N, int** v)
```

som returnerer antall “10-venner-tripler” i en 2D  $M \times N$  tabell  $v$ .

(Fortsettes på side 4.)

**5b** (vekt 10%)

Skisser en MPI implementasjon (vis bare de viktige detaljene), hvor initielt bare prosess 0 har verdiene til  $v$ .

**5c** (vekt 5%)

Utledd først en modell for tidsbruk av den parallelle implementasjonen, utledd så den assosierte funksjonen av *isoefficiency*. (Du kan nå anta  $M = N$ .)