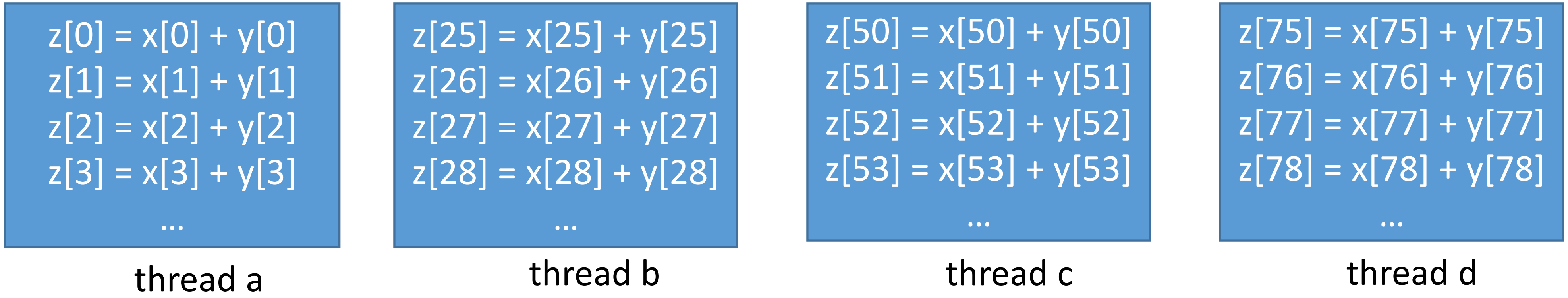
- Q1: 下面為OpenMP程式碼片段，OpenMP parallel region將產生數個thread，請說明每個thread的執行內容為何?

```
void v_add(double* x, double* y, double* z) {
  #pragma omp parallel
  {
    for(int i=0; i<ARRAY_SIZE; i++)
      z[i] = x[i] + y[i];
  }
}
```

- Q2: 下面為OpenMP程式碼片段，OpenMP parallel region將產生數個thread，請說明每個thread的執行內容為何?

```
void v_add(double* x, double* y, double* z) {
    #pragma omp parallel for
    for(int i=0; i<ARRAY_SIZE; i++)
      z[i] = x[i] + y[i];
}
```

- Q3: 請利用OpenMP改寫下面的程式碼片段，平行化for-loop的執行，同時以滿足下列的要求:
  - OpenMP parallel region會產生4個 work threads
  - 4個thread執行的運算內容如下:



z[0] = x[0] + y[0]
z[1] = x[1] + y[1]
z[2] = x[2] + y[2]
z[3] = x[3] + y[3]
…
thread a

z[25] = x[25] + y[25]
z[26] = x[26] + y[26]
z[27] = x[27] + y[27]
z[28] = x[28] + y[28]
…
thread b

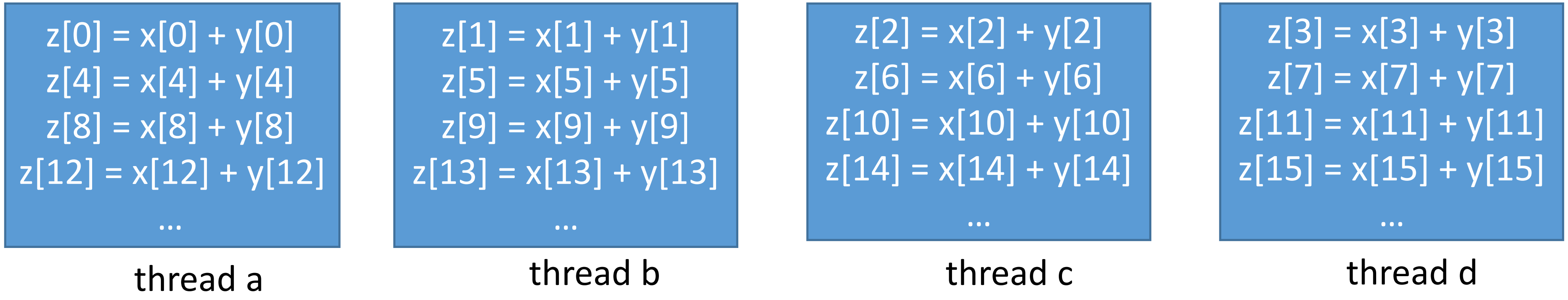
z[50] = x[50] + y[50]
z[51] = x[51] + y[51]
z[52] = x[52] + y[52]
z[53] = x[53] + y[53]
…
thread c

z[75] = x[75] + y[75]
z[76] = x[76] + y[76]
z[77] = x[77] + y[77]
z[78] = x[78] + y[78]
…
thread d

```
void v_add(double* x, double* y, double* z) {

  for(int i=0; i<100; i++)
      z[i] = x[i] + y[i];
}
```

- Q4: 請利用OpenMP改寫下面的程式碼片段，平行化for-loop的執行，同時以滿足下列的要求:
  - OpenMP parallel region會產生4個 work threads
  - 4個thread執行的運算內容如下:



z[0] = x[0] + y[0]
z[4] = x[4] + y[4]
z[8] = x[8] + y[8]
z[12] = x[12] + y[12]
…

thread a

z[1] = x[1] + y[1]
z[5] = x[5] + y[5]
z[9] = x[9] + y[9]
z[13] = x[13] + y[13]
…

thread b

z[2] = x[2] + y[2]
z[6] = x[6] + y[6]
z[10] = x[10] + y[10]
z[14] = x[14] + y[14]
…

thread c

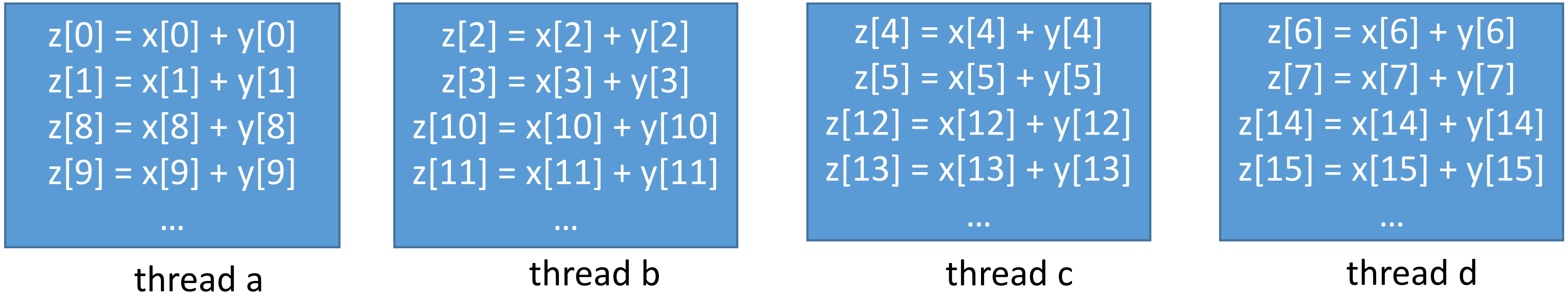
z[3] = x[3] + y[3]
z[7] = x[7] + y[7]
z[11] = x[11] + y[11]
z[15] = x[15] + y[15]
…

thread d

```
void v_add(double* x, double* y, double* z) {

  for(int i=0; i<100; i++)
      z[i] = x[i] + y[i];
}
```

- Q5: 請利用OpenMP改寫下面的程式碼片段，平行化for-loop的執行，同時以滿足下列的要求:
  - OpenMP parallel region會產生4個 work threads
  - 4個thread執行的運算內容如下:



z[0] = x[0] + y[0]
z[1] = x[1] + y[1]
z[8] = x[8] + y[8]
z[9] = x[9] + y[9]
…

thread a

z[2] = x[2] + y[2]
z[3] = x[3] + y[3]
z[10] = x[10] + y[10]
z[11] = x[11] + y[11]
…

thread b

z[4] = x[4] + y[4]
z[5] = x[5] + y[5]
z[12] = x[12] + y[12]
z[13] = x[13] + y[13]
…

thread c

z[6] = x[6] + y[6]
z[7] = x[7] + y[7]
z[14] = x[14] + y[14]
z[15] = x[15] + y[15]
…

thread d

```
void v_add(double* x, double* y, double* z) {

  for(int i=0; i<100; i++)
      z[i] = x[i] + y[i];
}
```

Please write down your answer in a WORD file and turn in to ECOURSE2.

- Deadline: December 19 (Tuesday), 24:00.