Pract 8

1.b blok 3: 448-479

blok 2: 160-191

blok 1: 128-159

blok 0: 96-127

2. Het aantal missers kunnen we berekenen met volgende formule: , dit is het aantal cacheblokken dat we nodig hebben en dus ook onze missers. Het aantal ingelezen bytes is het aantal toegangen\*4. Onze missrate verkrijgen we dan door nog eens te delen door het aantal toegangen: . Deze formule geldt voor blokgroottes kleiner of gelijk aan de optimale blokgrootte.

3. Bij een rowmajor is de temporele lokaliteit groot waardoor associativiteit de missrate niet zal beïnvloeden. De missrate hangt enkel af van de blokgrootte.

4. De geheugenlocaties volgen elkaar steeds op, we spreken van spatiale lokaliteit.

5. Onze missrate is nu 100%, dit komt omdat er steeds een andere geheugenlocatie wordt opgevraagd waardoor de inhoud van de cache steeds moet aangepast worden.

6. We krijgen het beste resultaat bij een FullyAssociative cache met blokgrootte 8. Dit geeft het beste resultaat omdat de geheugenlocaties gespreid liggen.

Deel 2

1. A en B zullen een lage missrate hebben aangezien zij rowmajor zijn en B een hoge missrate omdat deze columnmajor is. Als we dit controleren krijgen we 8/512 voor A en C en 288/512 voor B. Voor het de missrate van C hebben we enkel de reads geteld.

2. C heeft de beste temporele lokaliteit. We kunnen dit in de Java-code zien omdat die in de binnenste lus twee keer achter elkaar op hetzelfde adres aangeroepen wordt. 1 maal om de oude waarde uit te lezen en 1 maal om de nieuwe waarde op dezelfde plaats in het geheugen weg te schrijven

5. Door het in stukken te verdelen wordt de data van matrix B niet meer zuiver columnrow opgevraagd waardoor de temporele lokaliteit verhoogd en dus de missrate zal dalen.