De tweede programmeeropgave — Backtracking
Algoritmiek voorjaar 2007, Universiteit Leiden

De bedoeling van deze opgave is om op een M bij N "schaakbord"een serie zetten van een paard te genereren, zodat elk veld precies één (het startveld mogelijk twee, zie optie b. hieronder) keer bezocht wordt.
Een zettenserie van het paard waarbij alle $M * N$ velden van het bord één keer bezocht worden zullen we een "rit"noemen.

Beweging van een paard:

		8		1		
	7				2	
			P			
	6				3	
		5		4		

Een paard kan, uitgaande van plek P, de velden aangegeven met $1, 2, \ldots, 8$ in één beurt bereiken. Uiteraard zijn er minder dan acht mogelijkheden als het paard op de rand/in een hoek van het bord staat.

De opdracht bestaat uit twee delen.

1. Een **C++-programma** dat het volgende doet.

   De grootte van het bord, M (het aantal rijen) en N (het aantal kolommen), moet ingelezen worden, alsmede de coördinaten van het veld waar het paard moet starten. Dit mag gewoon met `cin`, maar er moet wel gecontroleerd worden dat het startveld binnen het bord valt. Vervolgens mag de gebruiker kiezen uit de volgende twee opties.

   a. een rit beginnend op het opgegeven veld wordt gefabriceerd en afgedrukt.

   b. een *gesloten* rit beginnend en eindigend op het opgegeven veld wordt geconstrueerd en afgedrukt.

Onder een *gesloten* rit die begint (en eindigt) op een gegeven veld verstaan we een rit waarbij het paard niet alleen alle $M * N$ velden precies één keer aandoet, maar waarbij het daarna ook weer in dat startveld terugkomt.

Een oplossing van het probleem (dus een rit op een M bij N bord) kunnen we weergeven door de velden te nummeren met $1, 2, \ldots, M * N$ zodat de waarden in de velden de volgorde aangeven waarin deze door het paard bezocht worden.

Voorbeeld 1. Een rit op het 3 bij 7 bord beginnend in het veld linksboven is:

1	14	17	20	11	8	5
16	19	12	3	6	21	10
13	2	15	18	9	4	7

In totaal zijn er trouwens 8 (alle niet-gesloten) ritten mogelijk op het 3 bij 7 bord met als startveld het veld linksboven.

Voorbeeld 2. Een gesloten rit op het 6 bij 6 bord, beginnend en eindigend in het veld linksboven is:

1	20	11	8	3	6
18	31	2	5	12	9
21	36	19	10	7	4
30	17	32	23	26	13
35	22	15	28	33	24
16	29	34	25	14	27

Opmerkingen:

- Het programma moet gebruik maken van de methode BACKTRACKING. Je mag zelf kiezen of je hierbij recursie gebruikt of niet.
- Een rit moet op het scherm getoond worden op de hierboven gegeven manier. Het kader is echter niet nodig, je kunt volstaan met alleen de getallen in matrixvorm. Als er geen (gesloten) rit mogelijk is moet dat gemeld worden.

- Voor het vinden van een rit met behulp van backtracking gaat men steeds in een vaste volgorde alle acht richtingen bekijken waarin het paard kan gaan, bijvoorbeeld gewoon in de volgorde $1, 2, \ldots, 8$.
  Een bekende heuristiek (vuistregel), reeds in 1823 bedacht door ene J.C. Warnsdorff, luidt dat het paard altijd eerst naar die velden geschoven moet worden die het moeilijkst bereikbaar zijn. Dat wil zeggen waarvoor er de minste mogelijkheden zijn om er te komen vanuit nog niet eerder betreden velden. In het algemeen lijken de hoekvelden bijvoorbeeld lastiger dan een veld in het midden van het bord, omdat je daar veel gemakkelijker in terecht kan komen (namelijk in het begin op acht manieren tegenover twee voor een hoekveld). Door nu het aantal nog niet bezochte plekken van waaruit een veld met één sprong te bereiken is bij te houden en na elke stap aan te passen, en de richtingen in volgorde van die toegankelijkheid te proberen, kun je sneller een rit vinden dan zonder gebruik van de heuristiek.

  Bij de optie onder a. moet de gebruiker kunnen kiezen tussen backtracking zonder deze heuristiek en backtracking met deze heuristiek.
- Er moeten drie verschillende memberfuncties geschreven worden bij een object bord, namelijk een voor optie a. zonder heuristiek, een voor optie a. met heuristiek en een voor optie b.. De laatste twee zijn (eenvoudige) aanpassingen van de eerste.
- Boven elke functie moet een commmentaarblokje komen met daarin een (zeer) korte beschrijving van wat de functie doet. Noem daarin ook de gebruikte parameters: geef hun betekenis en geef aan hoe ze eventueel veranderd worden door de functie. Geef bij memberfuncties ook aan wat deze met de membervariabelen van het object doen. Let ook op de layout (consequent inspringen) en op het overige commentaar bij de programmacode (alleen zinvol en kort commentaar).

2. Een getypt "**verslag**", waarin het volgende.

   - Leg duidelijk (maar niet al te lang) uit hoe de methode backtracking voor het vinden van een rit werkt en hoe de aanpassingen voor het vinden van een gesloten rit en het gebruik van de heuristiek zijn gedaan.
   - Ga zelf voor enige testgevallen het verschil na tussen backtracking met en zonder heuristiek. Doe van de experimenten verslag in het verslag.
   - Toon aan dat er nooit een gesloten rit gemaakt kan worden op een M bij N bord met $M * N$ oneven. Hint: kleur de velden als bij een schaakbord.
   - Eventuele extra informatie over de programmeeropdracht is te vinden op:
     `http://www.liacs.nl/home/jlaros/edu/alg.shtml`
     Voor meer informatie over het vak, laatste wijzigingen, etcetera, raadplege men:
     `http://www.liacs.nl/home/graaf/ALGO/algo2007.html`

Uiterste inleverdatum: vrijdag 13 april 2007. Voor elke week te laat inleveren gaat er een punt van het cijfer af. Het programma per e-mail sturen aan de hoofdnakijker: jlaros@liacs.nl. Listing en verslag moeten op papier worden ingeleverd en in de daartoe bestemde doos met opschrift Algoritmiek in de postkamer van Informatica (kamer 156) worden gedeponeerd. Vermeld overal duidelijk de namen van de makers.

Normering: verslag 2; commentaar en layout 1,5; modulaire opbouw en OOP 1,5; werking 5.