

Een terminalvenster is onder de meeste window managers gemiddeld circa 25 regels hoog en 80 tekens (kolommen) breed. Stel je het karakter op de middelste regel en de middelste kolom voor als coördinaat $(0, 0)$ – d.w.z. de *oorsprong* -- **van een discreet xy-grid**. Ga na dat mogelijke coördinaten in dit *grid* met stapgrootte 1 lopen van linksboven (circa -40, 12) t/m rechtsonder (circa 40, -12). Tel dit maar eens na in je favoriete terminalvenster van je VM. Een voorbeeld onder de Windows command line zie je in de geplote figuur hieronder.

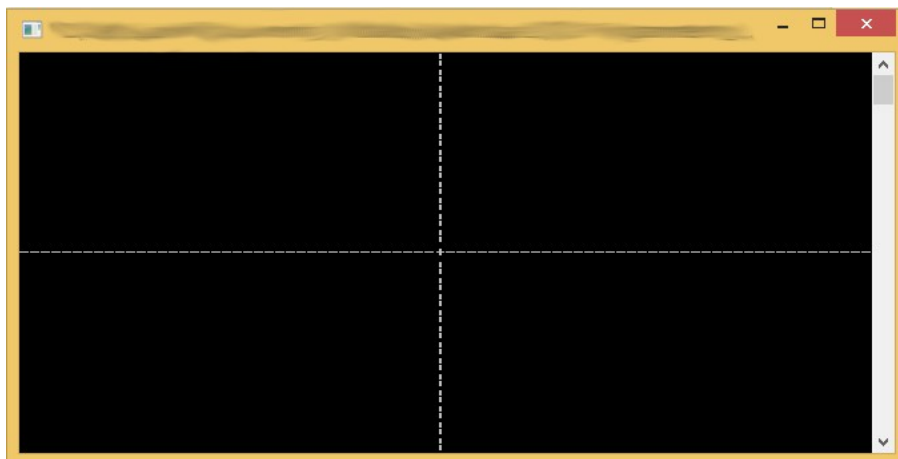
In deze opdracht zul je stapsgewijs het *grid* tonen in een terminalvenster (opgave A) om daar vervolgens een lijnstuk in te plotten op basis van 2 opgegeven punten (opgave B). Je bent bij deze opdracht in het voordeel als je al begrijpt hoe je moet werken met structs. Concepten als ‘point’ en ‘line’ lenen zich uitstekend voor het toepassen van structs. Deelopgave A kan zonder structs, onderdeel B is lastiger zonder structs te gebruiken ...

- A. Om het *grid* te tonen zul je op basis van een aantal vastgelegde grenswaarden een karakter willen printen. Neem deze grenswaarden als constanten in je code op en noem ze `MIN_X`, `MIN_Y`, `MAX_X` en `MAX_Y`. Let op: sommige van deze waarden zullen negatieve gehele getallen zijn (!)

Schrijf en demonstreer een functie die het beschreven *grid* op de console plot. Gebruik het volgende functieprototype:

```
void plot_grid();
```

Let erop dat je de rijen van `MAX_Y` t/m `MIN_Y` tekent (decrementeren), en dat je de kolommen van `MIN_X` t/m `MAX_X` toont (incrementeren). Demonstreer je functie door het volgende resultaat te reproduceren (neemt het gehele consolevenster in beslag, let op het plusje in het midden, voorbeeld in een Windows command line venster):



- B. Schrijf (en demonstreer) de functie `plot_line()`, die van een lijnstuk het gedeelte tekent dat in het venster zichtbaar is. Ga ervan uit dat coördinaten in dit stelsel altijd een **integer** waarde hebben. Een lijnstuk is gedefinieerd als de kortste weg tussen twee punten.

De beste aanpak is het ontwerpen van twee nieuwe datatypen: 'point' en 'line'. Een 'point' bestaat uit een **integer** x- en y-coördinaat. Een 'line' bestaat uit twee 'point's.

Je kunt het beste het volgende functieprototype gebruiken:

```
void plot_line(line *k);
```

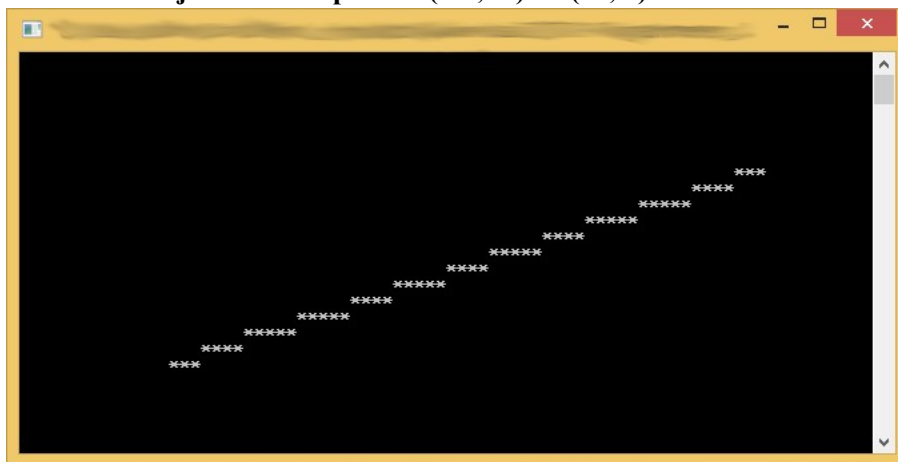
Als je niet weet hoe je met zo'n 'line' moet werken (of hoe je die definieert), dan wordt de lijst met argumenten langer. Geef in dat geval alle argumenten mee die een lijn beschrijven, bijvoorbeeld twee X- en Y-paren (4 integers in totaal). Je kunt de opdracht dan alsnog uitvoeren. Je bent ook niet verplicht een line-pointer (`line *`) te gebruiken, zoals hierboven. Dit is wel de meest efficiënte aanpak, maar het kan ook werken door simpelweg een 'line' mee te sturen ...

TIP(S):

- De formules voor het bepalen van de richtingscoëfficiënt en het snijpunt met de y-as uit twee gegeven punten op de lijn, om de 'a' en 'b' uit de lijnformule $y = a * x + b$ te berekenen, zijn hier van toepassing. Denk terug aan het vak 'Wiskundige Functies' van je wiskundedocent.
- Let erop dat je deze berekening nauwkeurig (floating point) uitvoert. Je krijgt immers kommagetallen als je gaat lopen delen. Na het uitvoeren van de wiskundige functie-stappen zul je echter wel moeten afronden om integer coördinaten te verkrijgen en om te bepalen of er op deze plek wel / geen 'sterretje' moet worden geplott.
- Je mag `<math.h>` includen. Voor het afronden van getallen is binnen deze library de functie `"double round (double x);"` beschikbaar.
- Je hebt misschien / waarschijnlijk functies als `MIN(a, b)` en `MAX(a, b)` nodig om de kleinste en de grootste van twee getallen te bepalen. Je kunt dan snel zien waar de 'uiterste grenzen' van de lijn liggen bij de weergave. Maak hier zelf macro's of functies voor.
- Je mag de output van deelopgave A combineren met de output van deze opgave, zodat je het *grid* ziet achter de geplote lijn. Zie de afbeelding van 'Test 4' hieronder.

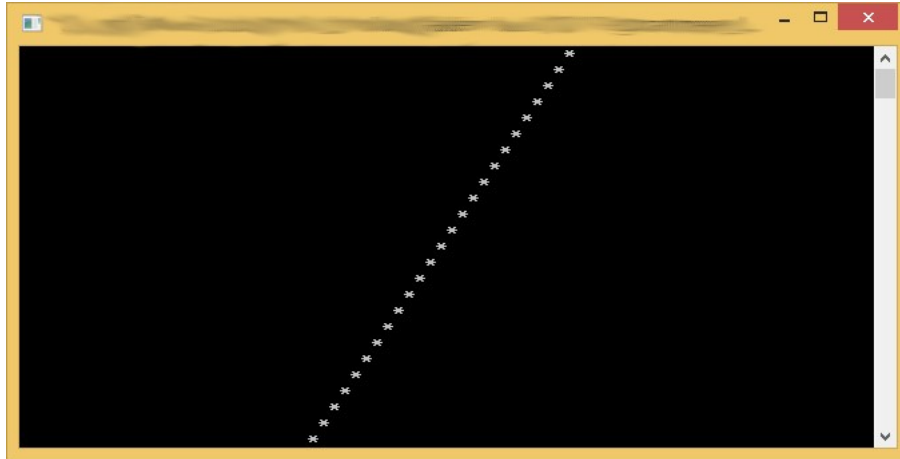
Demonstreer je functie door de volgende resultaten te reproduceren:

Test 1 – een lijn tussen de punten (-25, -7) en (30, 5):



Test 2 – een lijn tussen de punten (-233, 0) en (-45, 80). Dit lijnstuk ligt geheel buiten het grid en levert dus het lege grid / de lege terminal op (niet weergegeven, zie afbeelding bij 4A).

Test 3 – een lijn tussen de punten $(-200, -200)$ en $(200, 200)$, ofwel de vergelijking $y = x$:



Test 4 – een lijn tussen de punten $(-30, 20)$ en $(44, -20)$. Deze heeft een negatieve richtingscoëfficiënt en een herkenbaar stramien door de afronding (hier tevens inclusief *grid*):

