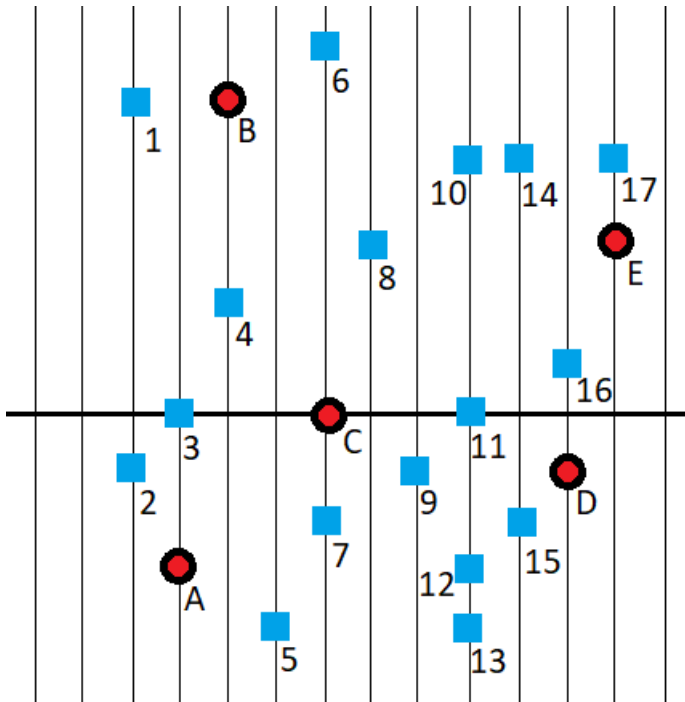


# Opgave 2. Meerijden

Deze opgave speelt zich af in het plaatsje Harkland. Dat heeft een bijzondere wegenstructuur; er loopt één hoofdweg van Oost naar West en er lopen allemaal evenwijdige wegen van Noord naar Zuid. Alle locaties hebben twee coördinaten, een OW- en een NZ-coördinaat; voor elke coördinaat  $c$  geldt  $-1000 < c < 1000$ .



Voor een excursie moeten deelnemers zich verzamelen bij één of meer pleinen, waar busjes ze kunnen oppikken. De busmaatschappij heeft een maximum aantal plaatsen per busje, en wil graag met zo weinig mogelijk busjes rijden. De organisatie wil dat de deelnemers zo weinig mogelijk hoeven reizen naar een opstapplaats. Jij moet bij dit probleem een aantal hulpprogramma's schrijven.

## Invoer

De eerste regel bestaat uit drie getallen  $p$ ,  $d$  en  $m$ , met  $p$  het aantal pleinen,  $d$  het aantal deelnemers en  $m$  het maximum aantal plaatsen per busje. De getallen worden gescheiden door spaties. Er geldt:

$$2 \leq p \leq 26$$

$$p < d \leq 99$$

$$1 < m \leq 50$$

Daarna volgen er  $p$  regels met elk twee getallen, gescheiden door spaties, waarop de coördinaten van de pleinen staan. Pleinen worden aangeduid met een hoofdletter, vanaf A op alfabetische volgorde.

Vervolgens volgen er  $d$  regels met elk twee getallen, gescheiden door spaties, waarop de coördinaten van de deelnemers staan. Deelnemers worden aangeduid met nummers, vanaf 1 olopend.

Voorbeeld (dat correspondeert met de afbeelding)

```
5 17 5
-3 -3
-2 5
0 0
5 -1
6 3
-4 5
-4 -1
-3 0
-2 2
-1 -4
0 6
0 -2
1 3
2 -1
3 4
3 0
3 -3
3 -4
4 4
4 -2
5 1
6 4
```

De invoergegevens zijn zo dat geen van de deelnemers op een plein woont, en het aantal pleinen is minstens zo groot als het aantal noodzakelijke busjes.

In het voorbeeld staan de locaties gesorteerd op eerste coördinaat; dat is voor de andere tests niet het geval!

### Overzicht:

In totaal kun je voor opgave 2 maximaal ongeveer 200 punten halen (van de in totaal ongeveer 400).

Opgave	Tijdlimiet	Testen	Per test	Totaal
2A	1	7	1	7
2B	1	7	3	21
2C	1	7	4	28
2D	2	7	5	35
2E	2	7	7	49
2F	5	5	ca. 12	ca. 60

## 2A: Aantal busjes

Schrijf een programma dat de informatie over Harkland inleest van standard input.

Het programma schrijft naar één regel van standard output het minimum aantal busjes dat nodig is om alle deelnemers aan de excursie mee te nemen.

Voorbeeld:

| 4

Toelichting: Met 3 busjes kunnen maximaal  $3 \times 5 = 15$  deelnemers worden meegenomen. Met 4 zijn dat er 20 en dat is wel genoeg.

## 2B: Iedereen naar hetzelfde plein

De eerste planner wil het simpel houden: Laat iedereen naar hetzelfde plein komen, dan komen ook alle busjes naar dat plein.

Schrijf een programma dat de informatie over Harkland inleest van standard input.

Je programma schrijft naar één regel van standard output de naam van het plein waar alle busjes moeten verzamelen. De gezamenlijke reisafstand voor alle deelnemers moet voor dit plein minimaal zijn; als die afstand voor verschillende pleinen hetzelfde is kies je voor het plein dat voorop komt in alfabetische volgorde (dus bij gelijke lengte naar B en D kies je de B).

Voorbeeld:

| C

Toelichting: De reisafstanden tot de verschillende pleinen zijn 172 voor A, 191 voor B, 94 voor C, 130 voor D en 173 voor E.

## 2C: Ondergrens

De tweede planner begint met de vraag wat de minimale totale reisafstand zou zijn als iedereen naar het dichtstbijzijnde plein zou reizen.

Schrijf een programma dat de informatie over Harkland inleest van standard input.

Je programma schrijft naar één regel van standard output de minimale reisafstand als elke deelnemers naar het dichtstbijzijnde plein gaat.

Voorbeeld

| 76

## 2D: Gretig

De derde planner maakt een indeling volgens het volgende algoritme.

Herhaal zolang niet alle deelnemers zijn ingedeeld:

    Zoek degene met de kortste reisafstand tot een plein waar een busje kan worden aangevuld. Zo nodig kies je de deelnemer met het laagste nummer en voor die deelnemer het eerste plein van de alfabetische volgorde.

    Vul het busje op het gevonden plein met de deelnemer. Als het busje nu vol is markeer je het als niet langer beschikbaar.

    Let erop dat je niet meer busjes dan nodig inzet! (zie opgave 2A)

In dit algoritme vertrekt van ieder plein hoogstens één busje.

Bekijk ook de toelichting hieronder waarin het algoritme voor het voorbeeld is uitgewerkt!

Schrijf een programma dat de informatie over Harkland inleest van standard input.

Het programma schrijft naar standard output een regel met daarop voor iedere deelnemer het plein waar hij naar toe moet gaan. Dat zijn dus  $d$  letters achter elkaar.

Voorbeeld

| AAACAACCCDCDEDDDE

Toelichting:

Achtereenvolgens deel je in:

17	in bus E	reisafstand 1	
7	in bus C	reisafstand 2	
16	in bus D	reisafstand 2	
3	in bus A	reisafstand 3	N.B. Er zijn nu vier bussen in gebruik
9	in bus C	reisafstand 3	
11	in bus C	reisafstand 3	
4	in bus C	reisafstand 4	(want bus B zou de vijfde bus zijn)
8	in bus C	reisafstand 4	(bus C is nu vol)
15	in bus D	reisafstand 4	
2	in bus A	reisafstand 5	
12	in bus D	reisafstand 6	
14	in bus D	reisafstand 6	
10	in bus D	reisafstand 7	(bus D is nu vol)
1	in bus A	reisafstand 9	
5	in bus A	reisafstand 9	
13	in bus E	reisafstand 10	
6	in bus A	reisafstand 12	(bus A is nu vol)

De totale reisafstand is zo 90.

## 2E: Eerst de lastigste

Weer een andere planner begint andersom: hij deelt telkens eerst degene van wie de reisafstand tot het dichtstbijzijnde plein maximaal is in en herhaalt dat proces tot alle deelnemers zijn ingedeeld. Ook in dit algoritme vertrekt van ieder plein hoogstens één busje.

Schrijf een programma dat de informatie over Harkland inleest van standard input.

Het programma schrijft naar standard output een regel met daarop voor iedere deelnemer het plein waar hij naar toe moet gaan. Dat zijn dus  $d$  letters achter elkaar.

Voorbeeld:

```
| AAABCCADDCBCCDDAD
```

Toelichting:

Achtereenvolgens deel je in:

1	in bus A	9	
10	in bus C	7	
13	in bus C	7	
6	in bus C	6	
12	in bus C	6	
14	in bus D	6	
2	in bus A	5	
5	in bus C	5	Bus C is nu vol; na deze plaatsing volgt een herberekening!
7	in bus A	8	
8	in bus D	8	
9	in bus D	5	
15	in bus D	4	
3	in bus A	3	
4	in bus B	3	Bus E vervalt nu, er zijn er vier in gebruik
17	in bus D	6	Bus D is nu vol; na deze plaatsing volgt een herberekening!
16	in bus A	12	Bus A is nu vol; na deze plaatsing volgt een herberekening!
11	in bus B	10	

De totale reisafstand is zo 110.

## 2F: Nog beter?

En nu ligt de bal bij jou: kun je een algoritme uitwerken dat het beter doet dan de hierboven beschreven versies? Het aantal busjes blijft hetzelfde, maar er mogen eventueel wel meer busjes van hetzelfde plein vertrekken.

Schrijf een programma dat de informatie over Harkland inleest van standard input.

Het programma schrijft naar standard output een regel met daarop voor iedere deelnemer het plein waar hij naar toe moet gaan. Dat zijn dus  $d$  letters achter elkaar.

De jury heeft een programma dat in alle testgevallen betere resultaten geeft dan de algoritmen bij 2B, 2D en 2E. Als jij het bij een testgeval beter doet dan deze drie algoritmen krijg je in ieder geval punten. De meestal onbereikbare score die hoort bij opgave 2C geeft 20 punten. Hoe dichter je programma daar bij zit, des te meer punten scoor je.

Voorbeeld:

```
|AAAACCCCCDDDDDEEDE
```

Toelichting:

De totale reisafstand bij deze uitvoer is 85.

Als jouw programma het beter doet dan een totale reisafstand van 90 krijg je punten. De ideale uitkomst (maar niet te realiseren) is 76. Met dit voorbeeld zou je  $20 \cdot \frac{90-85}{90-76} = 7,14$  punten halen.