

Uitwerking opdracht

Opdracht : *vierkant van pascal*
Weeknummer : *5*
Studentnummer : *s1060679*
Naam student : *Dion van den Berg*
Specialisatie : *MEDT*
Pogingnummer : *1*

1. Vraagstelling

Teken een vierkant van pascal met een opgegeven getal.

2. Specificatie

Invoer

De gebruiker voert een getal in voor de grote van het vierkant. Optioneel kan hij ook een getal invoeren die in het vierkant komt te staan.

Uitvoer

De gebruiker krijgt een grid van getallen te zien in de vorm van een vierkant.

Verband tussen in- en uitvoer

De uitvoer wordt gemaakt door de ingevoerde getallen.

Beperkingen

Het vierkant kan niet overweg met komma getallen

Voorbeelden (testscenario's)

Test 1

Invoer:

vierkant(3)

Uitvoer:

[[1, 1, 1], [1, 2, 3], [1, 3, 6]]

Test 2

Invoer:

Print(paden(10))

Uitvoer:

```
1 1 1 1 1 1 1 1 1 1
1 2 3 4 5 6 7 8 9 10
1 3 6 10 15 21 28 36 45 55
1 4 10 20 35 56 84 120 165 220
1 5 15 35 70 126 210 330 495 715
1 6 21 56 126 252 462 792 1287 2002
1 7 28 84 210 462 924 1716 3003 5005
1 8 36 120 330 792 1716 3432 6435 11440
1 9 45 165 495 1287 3003 6435 12870 24310
1 10 55 220 715 2002 5005 11440 24310 48620
```

3. Ontwerp

Hoe maak ik het rooster ?

Door het te printen met een alignment en een vaste width.

Hoe ga ik alles berekenen ?

Door telkens de vorige lijst te gebruiken om de getallen mee op te tellen.

Hoe laat ik alles uiteindelijk zien ?

in regels netjes onderverdeeld tot een vierkant ter grote van het laatst berekende getal van het vierkant..

4. Pseudocode

Functie vierkant (rijgrootte , getal)

Hoofdlijst

Lijst

Vul lijst 1 met het getal op

Loop rijgrootte aantal keer

Nieuwlijst = [getal]

Loop rijgrootte -1 keer

Nieuwlijst =(lijst x + 1) + nieuwlijst x

Voeg alles toe aan hoofdlijst

Maak van nieuwlijst lijst

Functie paden

Haal grootte op van het laatst ingevoegde getal

Loop door functie vierkant uitkomst heen en voeg dit toe aan een string inclusief enters

Format dit met een width(ter grootte van het laatst ingevoerde getal + 1) en alignment naar rechts

5. Code

```
def vierkant (rijgrootte, getal=1) -> [[]]:
    # maak lijstjes aan om het vierkant in op te slaan
    alles = []
    lijst = []

    # vul de eerste rij met het getal
    for y in range(rijgrootte):
        lijst.append(getal)

    # vul rij 2 met telkens het getal van links en boven in het grid
    # en voeg daarna de lijst toe aan de grote lijst met alle lijsten.
    for z in range(rijgrootte):
        newlijst = [getal]

        for x in range(rijgrootte-1):
            newlijst.append(lijst[x+1] + newlijst[x])
        alles.append(lijst)
        lijst = newlijst
    return alles

def paden(rijgrootte, getal=1) -> str:
    # haal de tekst op uit de vierkant functie
    tekst = ''
    vierkantje = vierkant(rijgrootte, getal)
    # haal op hoe lang het laatste getal is
    laatsteInLijst = vierkantje[rijgrootte - 1]
    p = len(str(laatsteInLijst[rijgrootte - 1])) + 1
    # loop door alle getallen heen en zet het allemaal op 1 grote regel,
    # gescheiden door de lengte +1 van het laatste getal p
    for x in vierkantje:
        for y in x:
            tekst += '{:{align}{width}}'.format(str(y), align='>', width=p)
        tekst += '\n'
    return tekst

# print de voorbeelden
print(vierkant(3))
vierkant(3, 100)
vierkant(4)

print(paden(3))
print(paden(3, 100))
print(paden(4))
print(paden(6))
print(paden(8))
print(paden(10))
```

6. Test

Test 1

1 1 1

1 2 3

1 3 6

100 100 100

100 200 300

100 300 600

1 1 1 1

1 2 3 4

1 3 6 10

1 4 10 20

1 1 1 1 1 1

1 2 3 4 5 6

1 3 6 10 15 21

1 4 10 20 35 56

1 5 15 35 70 126

1 6 21 56 126 252

1 1 1 1 1 1 1 1

1 2 3 4 5 6 7 8

1 3 6 10 15 21 28 36

1 4 10 20 35 56 84 120

1 5 15 35 70 126 210 330

1 6 21 56 126 252 462 792

1 7 28 84 210 462 924 1716

1 8 36 120 330 792 1716 3432

1 1 1 1 1 1 1 1 1 1

1 2 3 4 5 6 7 8 9 10

1 3 6 10 15 21 28 36 45 55

1 4 10 20 35 56 84 120 165 220

1 5 15 35 70 126 210 330 495 715

1 6 21 56 126 252 462 792 1287 2002

1 7 28 84 210 462 924 1716 3003 5005

1 8 36 120 330 792 1716 3432 6435 11440

1 9 45 165 495 1287 3003 6435 12870 24310

1 10 55 220 715 2002 5005 11440 24310 48620