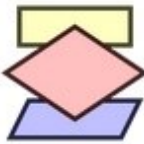




Oefen opdrachten Flowgorithm



Instructies

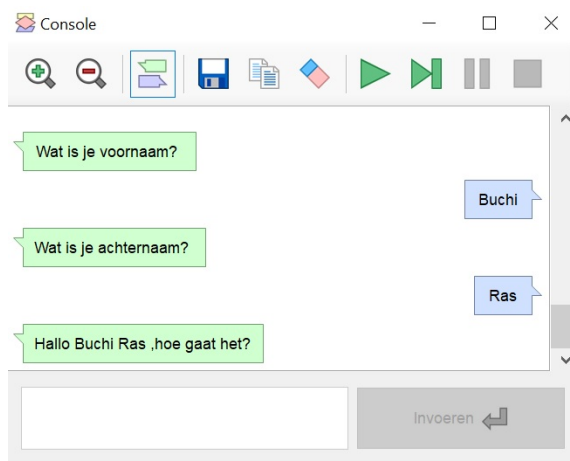
Afhankelijk van de opdrachten moet voor de **analyse** de volgende handelingen verrichten:

1. Bepaal wat de invoer moet zijn.
2. Bepaal wat de uitvoer moet zijn.
3. Bepaal de verwerking/het proces dat het programma moet uitvoeren.
4. Bepaal welke variabelen nodig zijn en welk type elk van deze moet hebben.
5. Bepaal de functies die nodig zijn. Beschrijf deze en bepaal de eventuele parameters die mee gegeven moeten worden en wat de mogelijke teruggeefwaarde is als er een nodig is.

Opdracht 1:

Maak een stroomdiagram die om je voornaam en achternaam vraagt en daarna een groet met je naam afdrukt.

Hiernaast zie je een voorbeeld van de mogelijke uitvoer.



Opdracht 2:

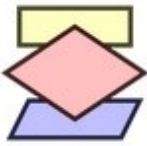
Maak een stroom diagram die vraagt om de naam en leeftijd van twee personen en daarna aangeeft welke van de twee ouder is of dat ze even oud zijn.

Hiernaast zie je een voorbeeld van de mogelijke uitvoer.





Oefen opdrachten Flowgorithm



Opdracht 3:

Maak een programma in Flowgorithm dat de gebruiker een simpel menu laat zien waar uit hij/zij een keuze kan maken. Het programma moet daarna tonen welke keuze de gebruiker geselecteerd heeft.

Hiernaast zie je een voorbeeld van de mogelijke uitvoer.

Console

Simpel menu voorbeeld

1 - keuze 1

2 - keuze 2

3 - keuze 3

Maak een keuze:

2

u heeft optie 2 gekozen.

Invoeren

Opdracht 4:

Maak een stroomdiagram die de eerste 10 uitkomsten van een door de gebruiker ingevoerde tafel op het beeldscherm laat zien.

Hiernaast zie je een voorbeeld van de mogelijke uitvoer.

Console

Welke tafel wilt u afgedrukt zien?

7

7 X 1 = 7

7 X 2 = 14

7 X 3 = 21

7 X 4 = 28

7 X 5 = 35

7 X 6 = 42

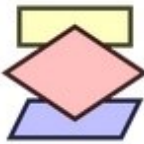
7 X 7 = 49

7 X 8 = 56

7 X 9 = 63

7 X 10 = 70

Invoeren



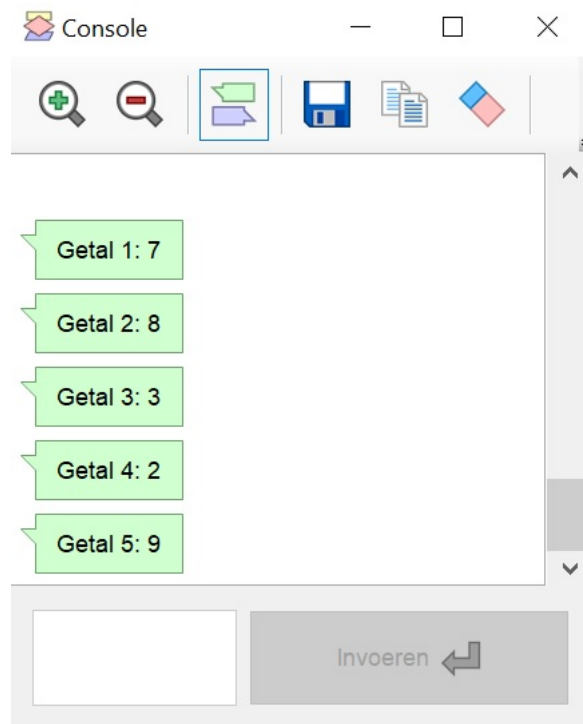
Oefen opdrachten Flowgorithm

Opdracht 5:

Voor deze opdracht heb je de ingebouwde functie **Random(n)** nodig. Deze functie produceert een willekeurige getal 0 en (n-1).

Maak een stroomdiagram dat 5 willekeurige getallen genereerd met een waarde tussen 0 en 10 en deze afdrukt in het console venster.

Hiernaast zie je een voorbeeld van de mogelijke uitvoer.

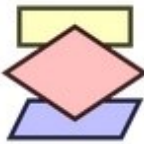


Opdracht 6:

Voor deze opdracht moeten een programma geschreven worden waarbij de gebruiker de namen en cijfers van twee verschillende leerlingen invoert in het programma. Waarop het programma daarna aangeeft of beide leerlingen hetzelfde cijfer hebben of de naam van de genen met het hoogste cijfer. Tevens moet het programma controleren of de leerlingen geen onvoldoende hebben. Lager dan een 5.5 wordt gezien als een onvoldoende.

Hieronder zie je voorbeelden van mogelijke in- en uitvoer van het programma:





Oefen opdrachten Flowgorithm

Opdracht 7:

Bij opdracht 6 moet je iedere keer als je nieuwe leerlingen wilt invoeren met nieuwe cijfers, het programma opnieuw opstarten.

Pas het programma zodanig aan dat dit niet meer noodzakelijk is.

Nadat het programma het resultaat heeft uitgevoerd moet het vragen voor een nieuwe leerling.

Wordt er echter geen naam ingevoerd (*lege string*) maar gewoon op <ENTER> gedrukt, dan moet het programma stoppen. Dit wordt aangegeven door de tekst "Tot ziens."

The screenshot shows the Flowgorithm interface with a list of messages and a text input field. The messages are as follows:

- Voer de 1e naam in:
- Voer het 1e cijfer in:
- Voer de 2e naam in:
- Voer het 2e cijfer in:
- Maria heeft het hoogste cijfer
- Pedro heeft een onvoldoende.
- Voer de 1e naam in:
- Voer het 1e cijfer in:
- Voer de 2e naam in:
- Voer het 2e cijfer in:
- Juan en Ana hebben even hoog cijfer.
- Voer de 1e naam in:
- Tot ziens.

The text input field contains the following entries:

- Maria
- 5.5
- Pedro
- 5.4
- Juan
- 6
- Ana
- 6

At the bottom, there is a text input field and an "Enter" button.

Opdracht 8A:

Maak een programma in Flowgorithm dat het spel hoger/lager speelt. Het programma moet een willekeurig (random) getal kiezen tussen 0 en 100. Daarna moet het programma de speler vragen het getal te raden binnen 10 beurten. Elke keer als de speler het getal niet heeft geraden geeft het programma een hint aan de speler door aan te geven dat het geheime getal hoger of lager is dan het getal dat de speler had geraden.

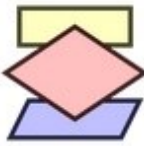
Als de 10 beurten om zijn moet het programma de speler vertellen wat het geheime getal was.

Als de speler het getal geraden heeft voor de 10 beurten om zijn moet het programma de speler feliciteren met zijn succes.

Na de 10 beurten of nadat het getal geraden is, moet het programma vragen of de speler nog een keer wilt spelen.



Oefen opdrachten Flowgorithm



Opdracht 8B:

Maak hetzelfde programma als bij opdracht 8A, maar nu moet de computer jouw getal op een slimme wijze vinden. Bedenk eerst een goede manier hoe de computer dit zou kunnen doen. Schrijf die stappen op onder elkaar in woorden.

Opdracht 9:

Maak een programma dat de gebruiker een willekeurig aantal positieve of negatieve getallen laat invoeren. Wanneer de gebruiker het cijfer nul invoert moet het programma stoppen en op het scherm tonen wat het grootste en kleinste getal was.

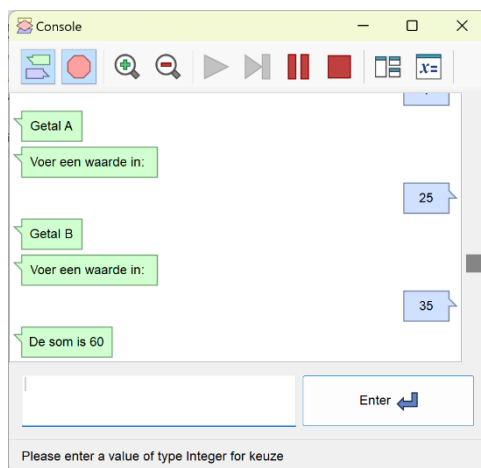
Opdracht 10:

Maak een programma dat de gebruiker een willekeurig aantal positieve getallen laat invoeren. Wanneer de gebruiker het cijfer nul invoert moet het programma het totaal en gemiddelde van alle getallen weergeven op het scherm.

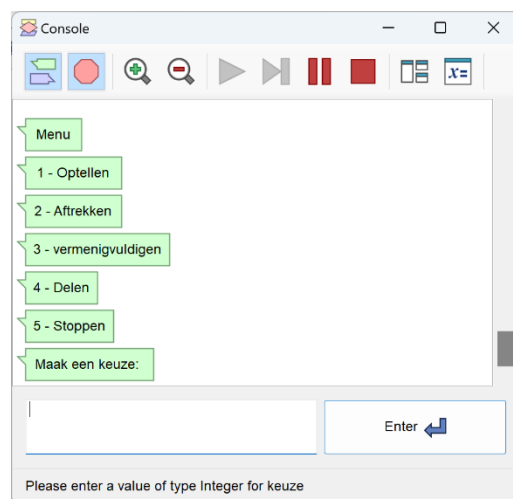
Opdracht 11:

Maak een programma dat een menu toont met de opties 1 voor optellen, 2 voor aftrekken, 3 voor vermenigvuldigen en 4 voor delen. Als de gebruiker het cijfer 5 intikt moet het programma stoppen.

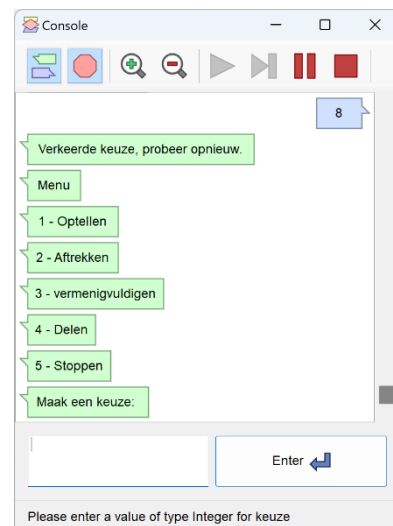
Er moet hier gebruik gemaakt worden van subroutines/functies.

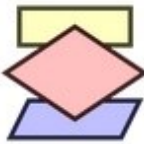


Wanneer de gebruiker een verkeerde keuze maakt moet het programma daar melding van geven.



Na elke berekening moet het programma het resultaat van de berekening op het scherm weergeven.





Oefen opdrachten Flowgorithm

Opdracht 12A:

Schrijf een programma dat een array met 100 elementen vult met willekeurige gehele getallen en daarna deze getallen weergeeft op het beeldscherm.

Opdracht 12B:

Breidt het programma van 12A uit. Laat een gebruiker een getal invullen waarna het programma moet controleren of het getal in de array voorkomt. Indien dat zo is moet het programma de index tonen waar het getal zich bevindt in de array. Zo niet moet het de gebruiker melden dat het getal niet in de lijst staat.

Opdracht 12C:

Pas het programma van 12B zodanig aan dat het aan geeft hoeveel keer het getal dat de gebruiker invoert voorkomt in de lijst en op welke index positie. Maak hier voor de maximale waarde van de random getallen van 100 naar 50.

Opdracht 13 Recursie:

Definitie Fibonacci-getallen

De Fibonacci-getallen zijn een reeks getallen in de volgende volgorde: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34... Het volgende getal wordt gevonden door de twee getallen ervoor bij elkaar op te tellen. De formule voor het berekenen van deze getallen is: $F(n) = F(n-1) + F(n-2)$.

Maak in Flowgorithm een recursief programma die een Fibonaccireeks van een door de gebruiker ingevoerde lengte afdruckt op het beeldscherm.

Opdracht 14A: Lijsten

Maak een programma dat de gebruiker een lijst van 10 willekeurige gehele getallen laat invoeren met een functie **VulLijst**."

Opdracht 14B

Maak bij het programma van 14A een functie **ToonLijst** welke de waarde van de elementen met hun index laat zien op het scherm.

Opdracht 14C

index	index	antw
5	50	55
10	45	55
15	40	55
20	35	55
25	30	55

Index	0	1	2	3	4	5	6	7	8	9
Waarde	5	10	15	20	25	30	35	40	45	50

Breidt het programma van **14A** en **14B** uit met een functie **Optellen**. Deze gaat de waarden van de eerste index positie optellen bij de laatste index positie en dan naar de volgende etc. en laat het resultaat van deze zien.