

Hoe maak je een programma geschikt voor de NIO?

Voor de meeste informatica opdrachten van de Nederlandse Informatica Olympiade (NIO) moet je de opdrachten op een bepaalde manier maken.

Dat is nodig voor:

- de drie wedstrijdronde van de NIO;
- voor de Internationale Informatica Olympiade;
- maar ook voor de CodeCup en voor Caia.

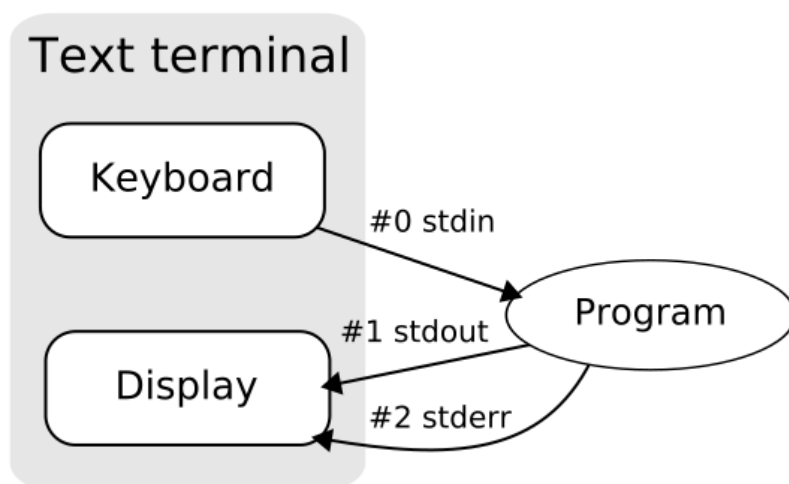
Als je wordt gevraagd een opdracht te programmeren waarbij je van stdin moet lezen en moet wegschrijven naar stdout, dan kan je de informatie in dit document gebruiken.

Wat zijn stdin en stdout?

Het zijn twee poorten, ook wel in het Engels standard streams genoemd. Standard streams zijn vooraf gedefinieerde invoer- en uitvoerkanalen (ook file-handles genoemd) tussen een computerprogramma en zijn omgeving.

stdin = standard input, dit regelt in gewone gevallen de invoer vanaf je toetsenbord naar je programma;

stdout = standard output, dit zorgt ervoor dat alles wat je afdrukt in je programma normaal gesproken op het scherm terecht komt.



Er bestaat ook nog de stream stderr: de zogenaamde standard error. Die kan je ook in je programma gebruiken om berichten naar het scherm te sturen.

Nu komt de grote truc. Als je een programma maakt voor NIO, dan gaan we jouw programma beoordelen. In dat geval laten we jouw programma geautomatiseerd uitvoeren. Wat normaal gesproken dan via het toetsenbord wordt ingevoerd voor jouw programma, sturen we dan via onze stdout naar de stdin van jouw programma. En alles wat jouw programma verstuurd naar stdout komt bij ons weer binnen op onze stdin. Alles wat jouw programma naar stderr stuurt, kunnen we

opvangen en in een logbestand voor je zetten. Dan kan je aan de informatie in dat bestand achteraf beoordelen wat je programma heeft gedaan. Het is aan jou welke informatie je daar naar toestuurt.

Op welke drie dingen moet je goed letten?

1. Doe precies wat in de opdracht staat en hou je aan de regels voor invoer en uitvoer. Belangrijk is dat je niet om invoer gaat vragen, maar dat je het ongevraagd inleest. Hetzelfde geldt voor de uitvoer: druk precies af wat gevraagd wordt en ga het niet opleuken met extra teksten.
2. Als je informatie afdrukt, sluit het dan altijd af met een newline. Anders kan de informatie niet ontvangen worden "aan de andere kant".
3. Je moet de informatie flushen. Dat betekent dat jouw programma de informatie, die je wilde afdrukken, dwingend doorstuurt.

Voorbeelden in verschillende programmeertalen

In de voorbeelden hieronder wordt het volgende probleem uitgewerkt. Schrijf een programma dat een integer inleest via stdin en het dubbele van die integer meteen afdrukt via stdout. Laat je programma stoppen als het getal -1 wordt ingevoerd.

C

```
#include

int main(void)
{
    int leesgetal, antwoord;
    while (1)
    {
        scanf("%d", &leesgetal);
        if (leesgetal == -1) break;
        antwoord = 2 * leesgetal;
        printf("%d\n", antwoord); // de newline is verplicht
        fflush(stdout);          // het flushen van het antwoord is noodzakelijk
    }
    return 0;
}
```

C++

```
#include
using namespace std;

int main(void)
{
    int leesgetal, antwoord;
    while (1)
    {
        cin >> leesgetal;
        if (leesgetal == -1) break;
        antwoord = 2 * leesgetal;
        cout << antwoord << endl; // de endl zorgt voor een newline en flusht
    }
    return 0;
}
```

Pascal

```
program sample;

var leesgetal, antwoord: integer;

begin
  while true do
    begin
      read(leesgetal);
      if leesgetal = -1 then halt(0);
      antwoord := 2 * leesgetal;
      writeln(antwoord); { gebruik writeln in plaats van write }
      flush(output);    { het flushen is noodzakelijk in Caia }
    end
  end.
end.
```

Python 2

```
#!/usr/bin/python
# Deze regel maakt dit bestand onder UNIX uitvoerbaar vanaf de prompt voor python 2
# Uiteraard moet python 2 daarvoor geïnstalleerd zijn!
# Je moet dit bestand eerst uitvoerbaar maken met: chmod +x sample.py

import sys                                # is verplicht voor de laatste twee regels

leesgetal = True

while leesgetal:
    leesgetal = int(raw_input())
    if leesgetal == -1:
        leesgetal = False
    else:
        antwoord = 2 * leesgetal
        sys.stdout.write(str(antwoord) + "\n") # de newline is verplicht
        sys.stdout.flush()                    # flushen is ook verplicht
```

Python 3

```
#!/usr/local/bin/python3
# Deze regel maakt dit bestand onder UNIX uitvoerbaar vanaf de prompt voor python 3
# Uiteraard moet python 3 daarvoor geïnstalleerd zijn!
# Je moet dit bestand eerst uitvoerbaar maken met: chmod +x sample.py

import sys                                # is verplicht voor de laatste twee regels

leesgetal = True

while leesgetal:
    leesgetal = int(input())
    if leesgetal == -1:
        leesgetal = False
    else:
        antwoord = 2 * leesgetal
        sys.stdout.write(str(antwoord) + "\n") # de newline is verplicht
        sys.stdout.flush()                    # flushen is ook verplicht
```

Java

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

public class Javasample
{
```

```
/**
 * @param args
 * @throws IOException
 * @throws NumberFormatException
 */
public static void main(String[] args) throws NumberFormatException, IOException
{
    BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));
    String line;
    while ((line = reader.readLine()) != null)
    {
        int number = Integer.parseInt(line);
        if (number == -1) break;
        System.out.println(Integer.toString(number * 2));
    }
}
```