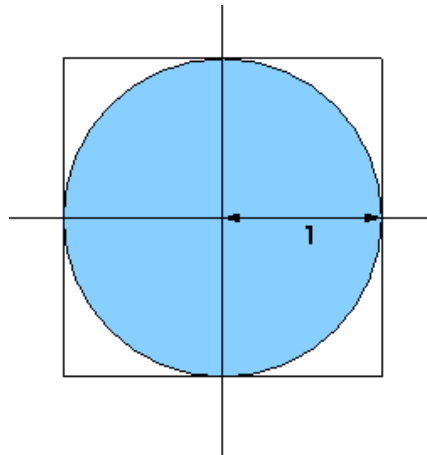


Opgave 1: Benaderen van Pi

Benaderen van Pi

De wiskundige constante Pi is de verhouding tussen de omtrek en diameter van een cirkel. In deze eerste opgave moet je een C programma schrijven dat een benadering van Pi berekent.

Je kan de waarde van Pi benaderen met behulp van de volgende methode. Neem een vierkant met daarin een cirkel met straal 1 zoals in het onderstaande plaatje.



Verhouding oppervlak cirkel en vierkant

De verhouding tussen het oppervlak van de cirkel en het oppervlak van het vierkant is:

$$\frac{\text{oppervlak cirkel}}{\text{oppervlak vierkant}} = \frac{\pi * r^2}{(2 * r)^2} = \pi / 4$$

Als we nu willekeurig een reeks punten kiezen in het vierkant dan is de verhouding tussen het aantal punten dat binnen de cirkel valt en het totaal aantal punten een benadering van de verhouding tussen het oppervlak de cirkel en het vierkant, oftewel $\pi/4$.

Stel dat van de 1000 punten er 784 binnen de cirkel vallen. Dan is de ratio $777/1000 = 0.777$.

$$\frac{\pi}{4} = 0.777 \Rightarrow \pi \approx 4 * 0.777 = 3.108$$

Hoe meer willekeurige punten er gekozen worden, hoe nauwkeuriger de benadering van Pi zal zijn.

De opdracht

Schrijf een C programma dat met de bovenstaande methode een benadering van Pi berekent. Aangezien het om de verhouding tussen de oppervlakken gaat kun je ook besluiten een kwart van de cirkel te gebruiken ipv de hele cirkel.

Het is voor een computer niet eenvoudig om efficient volledig random getallen te genereren. In de praktijk gebruik je een pseudo-random getalsgenerator zoals `rand()`. Lees de manual page zorgvuldig zodat je de functie correct gebruikt. Manual pages zijn te vinden met het commando `man`. In dit geval `man rand` of `man -s 3 rand` op sommige systemen. Ook nuttig is de manual page van `man` zelf: `man man`.

Je moet het aantal punten dat gebruikt wordt op kunnen geven via de command line bij de aanroep. Als er geen getal of een ongeldig getal meegeven wordt moet je programma dit herkennen en stoppen. De argumenten worden doorgegeven in een array van pointers naar strings, en natuurlijk wordt de grootte van de array ook meegegeven. Het prototype van `main` is:

```
int main(int argc, char *argv[])
```

Getallen op de command line worden ook doorgegeven als strings. Als je een argument als integer wilt gebruiken moet je deze eerst converteren met een functie als `atoi()`. Vergeet niet dat `argc` de programma naam ook meetelt. Dus als je je programma uitvoert met `./pi 1000`, zal `argc` de waarde 2 hebben, `argv[0]` de string `./pi` bevatten en `argv[1]` de string `"1000"`.

Gebruik voor het compileren een Makefile. Je kan de Makefile van de "even opstarten" sectie van de website kopiëren en aanpassen of er zelf een schrijven. Een korte introductie over Makefiles vind je hier: <http://staff.science.uva.nl/~andy/ProgC/makefiles.pdf>

Regels voor het inleveren

Omdat het nakijken van al de opgaven veel werk is, zijn de regels voor het inleveren zeer strict. Dit maakt het mogelijk om de ingeleverde opgaven automatisch te verwerken. Als je opgave niet aan de regels voldoet kan je dat punten kosten. Dat is natuurlijk zonde, lees daarom de regels over het inleveren zorgvuldig en stel vragen als het onduidelijk is. Het document met de regels voor het inleveren vind je hier: <http://staff.science.uva.nl/~andy/ProgC/inleveren.pdf>