Inlupp 1: Utskrifter och loopar

Imperativ Programmering DT501G, HT 2025
24 november 2025

I den här uppgiften kommer du att bekanta dig med och få övning i den nödvändigaste syntaxen i C och få övning i grundläggande hantering av variabler och flödeskontroll, samt input och output från program.

Se till att du har skapat en katalogstruktur enligt instruktionerna i inlupp 0. Filerna för den här inluppen ska alltså använda samma katalogstruktur som du skapade i inlupp 0, och ligga som `inlupp1/task_1.c`, `inlupp1/task_2.c`, och så vidare. Svaren på frågorna som är markerade i marginalen ska lämnas in i filen `inlupp1/svar.txt`.

Tänk också på att formatera dina utskrifter enligt det som står i instruktionerna. Står det att ditt program ska skriva ut `Area 3.14` så ska det också vara så, och inte `Arean är 3.14 m2` eller någon annan variant. (Om de inte gör det kommer kanske inte den automatiska testningen att fungera, vilket antagligen leder till mer arbete både för dig och den som rättar dina uppgifter.)

Det finns en uppgift som är markerad veckans mästarprov. Den är inte frivillig, och måste alltså lämnas in.

Lärandemål

Målet med de här uppgifterna är att du ska lära dig behärska följande moment:

- grundläggande text-inmatning och utmatning (till och från `stdin` och `stdout`),
- läsa och skriva data i variabler (heltal och flyttal),
- kommentarer i programkod,
- grundläggande flödeskontroll (`if`/`else` etc),
- algoritmiskt tänkande: att kunna skriva en algoritm för att lösa ett mindre problem,
- förstå hur tal representeras i en binär dator.

Uppgifter

1.1 Skriv ditt namn

Skriv ett C-program som skriver ut ditt namn, din e-postadress och en hobby på `stdout` enligt följande format (med varje post på en egen rad).

Till exempel:

```
Martin Magnusson
martin.magnusson@oru.se
Spela ukulele
```

1.2 Räkna med cirklar (1)

Skriv ett C-program som beräknar arean och omkretsen av en cirkel med radien $r = 1$, och skriver ut resultatet.

Du ska använda ekvationerna nedan för din beräkning, även om resultatet alltid blir samma. (Det är alltså inte en lösning att räkna ut svaret i förväg och skriva ut en fördefinierad textsträng!) Du kan approximera π med 3.1415 för den här uppgiften.

$$A = \pi r^2 \qquad O = 2\pi r$$

Skriv ut resultatet på två rader, enligt nedan. (Antal decimaler i utskriften spelar ingen roll i det här fallet, så länge det är minst två.)

```
Area 3.14
Omkrets 6.28
```

1.3 Räkna med cirklar (2)

Skriv ett C-program som beräknar arean och omkretsen av en cirkel som ovan, men som läser in radien från användaren. (Nu kommer du alltså att behöva använda en funktion även för att läsa input, t ex `scanf`, utöver `printf`.)

```
$ ./task_3
Radie? 10
Area 314.15
Omkrets 62.83
```

1.4 Loop

Skriv ett program som, med hjälp av en `for`-loop, skriver ut alla heltal från och med 1 till och med 20, separerade med mellanslag.

1.5 Loop/2

Skriv ett program som, med hjälp av en `for`-loop, skriver ut alla tal från och med 1 till och med 20 som är jämnt delbara med antingen 2 eller 5. Separera talen med mellanslag. (Här behöver du använda modulo-operatorn `%` samt OR-operatorn `||`.)

1.6 Loop de loop

Skriv ett program som tar två heltalsargument, ett för timmar och ett för minuter, och skriver ut klockslaget var tionde minut från kl 00:00 och fram till det angivna klockslaget. Läs in var sitt heltal för timmar och minuter med `scanf`.

```
$ ./task_6
Timme och minut? 1 15
00:00
00:10
00:20
00:30
00:40
00:50
01:00
01:10
```

För att skriva ut tal tvåsiffrigt (alltså ”02” i stället för ”2”) använder du specifikationen `%02d` till `printf`. (Se även kapitel 2.4.)

Du ska använda nästlade loopar i din lösning. I den yttre loopen itererar du lämpligen från 0 till timmen `h` som användaren angett. I den inre loopen ska du i stället iterera över minuter. Antingen hela vägen från 0 till 50, eller – om du är framme vid timme `h` – bara fram till minuten `m` som användaren angett. Här kan det vara smidigt att lägga in ett **`break`** på ett lämpligt ställe.

Du behöver inte hantera fall där någon skriver in en timme som är större än 23 eller ett minutvärde som är större än 59, eller skriver in något annat än siffror.

1.7 do

Skriv ett program som gör samma sak som i uppgift 1.6, men med **`do`**-loopar i stället.

1.8 while

Skriv ett program som gör samma sak som i uppgift 1.6, men med **`while`**-loopar i stället.

1.9 Slumptal

Skriv ett program som simulerar ett tärningskast (med en sex-sidig tärning). Varje gång programmet körs ska det helt enkelt skriva ut ett slumpvis valt tal mellan 1 och 6.

För att generera slumptal kan du använda funktionen `rand` som du får tillgång till med **`#include`** `<stdlib.h>`.[1] Funktionen `rand` returnerar ett slumpvis valt heltal mellan 0 och konstanten `RAND_MAX`, som är lite olika i olika C-implementationer. För att få ett värde mellan 1 och 6 kan du använda modulo-operatorn `%`. (Ett heltal – vilket som helst – modulo 6, blir ju alltid mellan 0 och 5!) För att få olika slumpvärden behöver du också anropa `srand`, som sätter ett ”slumpfrö”. För att få olika slumpvärden varje gång behöver du också välja frö från en källa som ändrar sig; t.ex. klockan. Du kan skicka `time(NULL)` som argument till `rand` får att få ett nytt frö varje sekund. Du får tillgång till `time` genom att inkludera standard-headern `time.h`.

[1] Du kan läsa mer om `rand` och andra funktioner från standardbiblioteken på The C Library Reference Guide.

1.10 Max3

Skriv ett program som läser in tre tal (med `scanf`), och som skriver ut det största av de tre talen. Tänk på att programmet ska göra rätt även om ett eller flera av talen är samma.

Veckans mästarprov

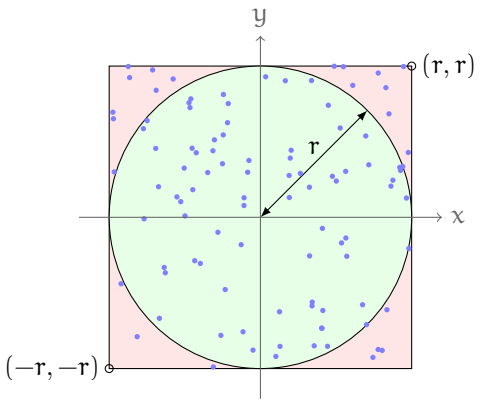
1.11 Räkna med cirklar (3)

Skriv ett C-program som beräknar arean av en cirkel, men nu med hjälp av så kallad Monte Carlo-integrering. Precis som ovan ska du läsa in radien från användaren.

Monte Carlo-integrering är ett sätt att numeriskt beräkna en integral (=area), som är användbar för väldigt knöliga funktioner. Arean av en cirkel är visserligen inte knölig, men med en så pass lätt funktion är det lätt att kontrollera att man har gjort rätt.

För att använda Monte Carlo-integrering så behöver du först bestämma ett intervall som du ska integrera över. För att få med hela cirkelns area behöver intervallet vara minst så stort som cirkeln. Ett vettigt intervall i det här fallet är då den kvadrat som innesluter cirkeln. Om du har en cirkel med mittpunkt $(0, 0)$ och radie $\mathtt{r} = 1$ så får du då en kvadrat som går mellan hörnen $(-1, -1)$ och $(1, 1)$.

Figuren nedan illustrerar det hela. Funktionen som vi vill integrera har värdet 1 i det gröna intervallet och 0 överallt annars. Intervallet som vi integrerar över (det röda) måste täcka minst det gröna intervallet, men kan annars vara i princip hur stort som helst.



Algoritmen för Monte Carlo-integrering ser ut så här:

1. Initiera en variabel för totalsumma: $\mathtt{S} \leftarrow 0$.
2. Initiera en variabel för skattning av arean: $\mathtt{A}$.
3. Initiera en räknare: $\mathtt{i} \leftarrow 0$.
4. Iterera följande tills du tror dig ha uppnått tillräcklig noggrannhet.
   (a) Stega upp räknaren: $\mathtt{i} \leftarrow \mathtt{i} + 1$.
   (b) Generera ett slumptal inom intervallet. (En blå punkt i bilden ovan.) För ett kvadratiskt intervall behöver du generera en tvådimensionell koordinat, det vill säga ett slumpvärde för $\mathtt{x}$ och ett för $\mathtt{y}$. Tänk på att för ett intervall som den röda kvadraten i figuren här måste koordinaterna ligga på intervallet $[-\mathtt{r}, +\mathtt{r}]$.
   Här ska du alltså skapa slumpvis valda flyttal i stället för heltal, som i den förra uppgiften. För att få ett värde mellan 0 och 1 kan du dividera

talet du får från `rand` med konstanten `RAND_MAX`. Men, eftersom de båda talen är heltal kommer det att bli heltalsdivision om du bara skriver `rand()/RAND_MAX`, och det är ju inte det du vill ha. Därför behöver du använda explicit typomvandling (cast) för att göra om minst ett av heltalen till flyttal: t ex med `(double)(rand())/RAND_MAX`.

(Beräkningen ovan ger ett slumptal mellan 0 och 1. Men hur gör man för att få ett tal mellan $-r$ och $+r$?)

(c) Beräkna värdet av funktionen som ska integreras för den genererade punkten. I vårt fall har funktionen värdet 1 om punkten ligger innanför cirkeln, och 0 annars. Här måste du alltså kolla om din slumpvis valda punkt (x, y) ligger innanför cirkeln; med andra ord, om $\sqrt{x^2 + y^2} \leqslant r$, eller motsvarande (för att slippa beräkna en kvadratrot): $x^2 + y^2 \leqslant r^2$.

(d) Multiplicera värdet du fick (0 eller 1) med arean för integreringsintervallet (det vill säga kvadraten); alltså med $(2r)^2$.

(e) Addera värdet till totalsumman som du räknar upp.

(f) Ändra den nuvarande uppskattningen av arean till $A \leftarrow S/i$.

5. Returnera A.

Vad innebär det då att iterera ”tillräckligt” länge, i steg 4 ovan? Dels kan du ju sluta när i har blivit för stort (alltså när du har loopat mer än ett visst antal gånger), men du ska också sätta ett stoppkriterium som beror på hur mycket A har ändrats. När den uppskattade arean bara ändrar sig med någon decimal långt bort så kommer den förmodligen inte att ändras så mycket mer även om du itererar längre i loopen. För att hålla koll på det behöver du en variabel B som kommer ihåg B från förra varvet i loopen, så att du kan jämföra om $|A - B| < \epsilon$, där du sätter ϵ till ett litet värde.

Du kan vilja använda någon funktion från biblioteket `math.h` i den här uppgiften. Funktionen `sqrt` därifrån beräknar kvadratroten av ett tal, och där finns även funktionen `fabs` som är praktisk för att få absolutvärdet av ett flyttal, och `pow` för att beräkna potenser. Förutom att inkludera `math.h` behöver du också länka med math-biblioteket. Vi kommer att prata mer om det lite senare i kursen, men just nu räcker det med att veta att du måste lägga till `-lm` på kommandoraden när du ska kompilera och bygga ditt program. Alltså: `gcc task_11.c -Wall -std=c99 -lm -otask_11`

Tanken med den här uppgiften är att öva mer på loopar, med mer avancerat stoppkriterium än en fast gräns, samt att prova på slumptalsgenerering.

Fråga 1 Hur länge behöver du iterera innan du får ungefär rätt värde på arean när $r = 1$ respektive $r = 2$?

Fråga 2 Hur många decimalers noggrannhet får du?

Fråga 3 Vad har du använt för stoppkriterium? Är det tillräckligt?

1.12 Binära talrepresentationer

Fråga 4 Skriv följande tal i binär form, som en `unsigned char` (alltså utan tvåkomplement). Något av talen går inte att representera på det här sättet. Ange vilket, och varför.

Fråga 5 Skriv talen i binär form, som en `signed char` (alltså med tvåkomplement). Något av talen går inte att representera på det här sättet. Ange vilket, och varför.

- 0
- 8
- 10

- 130
- -126