

Maskinorienterad programmering

Inlämningsuppgift

Detta PM innehåller anvisningar om genomförande av en inlämningsuppgift som ska redovisas innan laboration 4 påbörjas.

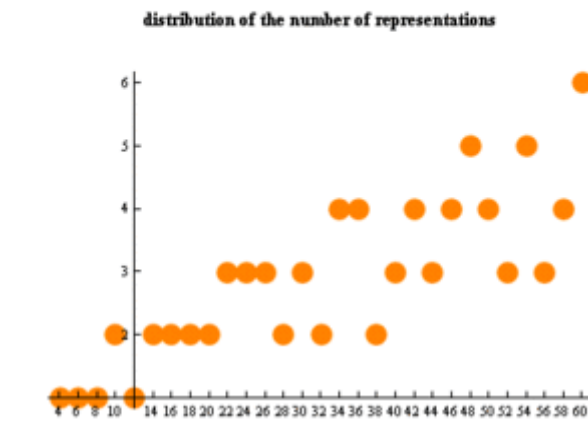
Goldbach hypotes

Goldbach's conjecture is one of the oldest unsolved problems in number theory and in all of mathematics. It states:

Every even integer greater than 2 can be expressed as the sum of two primes.

Such a number is called a Goldbach number. Expressing a given even number as a sum of two primes is called a Goldbach partition of the number.

...
(52 = 5 + 47, 52 = 11 + 41, 52 = 23 + 29)
(54 = 7 + 47, 54 = 11 + 43, 54 = 13 + 41, 54 = 17 + 37, 54 = 23 + 31)
(56 = 3 + 53, 56 = 13 + 43, 56 = 19 + 37)
(58 = 5 + 53, 58 = 11 + 47, 58 = 17 + 41, 58 = 29 + 29)
(60 = 7 + 53, 60 = 13 + 47, 60 = 17 + 43, 60 = 19 + 41, 60 = 23 + 37,



The number of ways an even number can be represented
as the sum of two primes

(Wikipedia)

Namn och personnummer med bläck.

Personnummer

Namn (textat)

Följande fylls i av laborationshandledare efter godkänd redovisning.

Godkännande av inlämningsuppgift	
Datum	Laborationshandledares underskrift

Med denna uppgift får du en första övning i att kompilera och köra ett C-program. Du får också träna på att skriva repetitionssater, *if*-satser och en enkel funktion. Dessutom får du utnyttja standardfunktioner för att göra in- och utmatning av data.

Förberedelser

Läs noga igenom uppgifterna så att du förstår vad du skall göra och hur problemen skall lösas. Gör en skiss av hur programmen skall se ut och skriv programkoden.

Programmeringsmiljö

Du får använda valfri programutvecklingsmiljö men vi rekommenderar *CodeLite*.

Uppgift 1

Ett primtal är ett heltal >1 som inte går att dela jämnt med några andra tal än med 1 och med sig själv. Uppgiften är att skriva ett C-program som läser in positiva heltal från tangentbordet och som undersöker om talen är primtal eller inte.

Ditt program *måste* bestå av två funktioner: `main` och `is_prime`. Dessa skall ligga i *var sin* fil. Funktionen `is_prime` skall undersöka om ett visst tal n är ett primtal eller inte. Talet n som skall undersökas skall ges som parameter till funktionen. Som resultat skall funktionen `is_prime` ge värdet 1 om det talet n är ett primtal och värdet 0 om talet n inte är ett primtal. För att undersöka om talet n är ett primtal kan man inne i funktionen `is_prime` direkt använda definitionen av primtal. Man kan alltså helt enkelt försöka att dividera talet n med alla andra tal som är större än 1 men mindre än n . Om divisionen inte går jämt ut för något sådant annat tal är n ett primtal.

I funktionen `main` skall man läsa in ett tal i taget från tangentbordet. För varje tal skall man anropa funktionen `is_prime` för att undersöka om det inlästa talet är ett primtal eller inte. Om det inlästa talet är ett primtal skall `main` skriva ut texten *Primtal* annars skall texten *Inte primtal* skrivas ut. Programmet skall vara så utformat att det avslutas om användaren matar in ett tal som är mindre än eller lika med 1.

Observera att funktionen `is_prime` *inte* skall läsa från tangentbordet och att den inte heller skall producera några utskrifter.

Provkör noga detta program och förvissa dig om att det fungerar innan du går vidare till nästa uppgift.

Uppgift 2

Den s.k. Goldbach hypotes säger att varje jämnt naturligt tal (utom 2) går att skriva som summan av två primtal. Det gäller t.ex. att $4=2+2$, $6=3+3$, $8=5+3$, $10=5+5$ och $12=5+7$. Hypotesen ställdes upp i mitten av 1700-talet och anses allmänt vara sann. Den är dock ännu ej bevisad.

Uppgiften är nu att skriva ett C-program som upprepade gånger frågar efter ett tal från tangentbordet. Om talet är jämnt skall programmet skriva ut alla möjliga kombinationer av två primtal som bildar talets summa. Varje kombination av primtal skall endast skrivas ut en gång. Om t.ex. talet 14 läses in, så skall programmet endast ge ett av primtalsparen (3,11) och (11,3) som utdata. Om ett inläst tal är udda skall programmet ge en felutskrift. Körningen skall avslutas om man matar in ett tal som är mindre än eller lika med 2. Ett exempel på hur det kan se ut när man kör programmet visas nedan:

```
Ge ett tal. Avsluta med ett tal <= 2: 14
3+11
```

7+7

Ge ett tal. Avsluta med ett tal ≤ 2 : 100

3+97

11+89

17+83

29+71

41+59

47+53

Ge ett tal. Avsluta med ett tal ≤ 2 : 897

Talet är udda!

Ge ett tal. Avsluta med ett tal ≤ 2 : 1

Programmet skall naturligtvis ha en funktion `main`. I denna skall all inläsning och utskrift ske. När ett tal `t` har lästs in skall man i `main` undersöka alla kombinationer av talen `n1` och `n2` så att $n1+n2=t$. Om både `n1` och `n2` är primtal har man funnit en lösning. För att undersöka om `n1` och `n2` är primtal skall `main` anropa funktionen `is_prime` från uppgift 1. Denna funktion skall fortfarande finnas i en egen fil.

Godkännande

När programmet i uppgift 2 fungerar skall det visas upp för en handledare för godkännande. För att uppgiften skall bli godkänd räcker det inte med att programmet fungerar. Det måste också vara skrivet på ett snyggt och begripligt sätt. Programraderna skall t.ex. indenteras (dras in) på det sätt som lärs ut i kursen.