Programmering i matematikundervisningen

Ulrica Dahlberg, NCM

NCM Nationellt centrum för matematikutbildning

GÖTEBORGS UNIVERSITET

Ändringar i kursplanen

- Lågstadiet, åk 1-3 Algebra

Hur entydiga stegvisa instruktioner kan konstrueras, beskrivas och följas som grund för programmering. Symbolers användning vid stegvisa instruktioner.

- Mellanstadiet, åk 4-6 Algebra

Hur algoritmer kan skapas och användas vid programmering. Programmering i visuella programmeringsmiljöer.

Ändringar i kursplanen

- Högstadiet, åk 7-9 Algebra

Hur algoritmer kan skapas och användas vid programmering. Programmering i olika programmeringsmiljöer.

- Högstadiet, åk 7-9 Problemlösning

Hur algoritmer kan skapas, testas och förbättras vid programmering för matematisk problemlösning.

Ändringar i kursplanen

- Gymnasiet Kurs 1c, 2c, 3b, 3c, 4, 5, specialisering; Problemlösning

Strategier för matematisk problemlösning inklusive modellering av olika situationer, såväl med som utan digitala verktyg och programmering.

Lite historia …

- Slutet av 60-talet började man undervisa i programmering
- Sedan dess olika grad och form av undervisning
- Papert utvecklade LOGO på 70-talet

Varför?

Vilket språk?

Nationellt centrum för matematikutbildning

GÖTEBORGS UNIVERSITET

Modul: Matematikundervisning med digitala verktyg II

NCM Nationellt centrum för matematikutbildning

GÖTEBORGS UNIVERSITET

Projekten är ett samarbete mellan NCM och

- Lärargrupp
- Aalborg universitet
  Morten Misfeldt
- Malmö högskola
  Ulrika Ryan
- Luleå tekniska univ.
  Maria Johansson
- Uppsala universitet
  Lennart Rolandsson

Syfte och mål

Lärare ska

- få möjlighet att uppmärksamma sin egen undervisning och specifikt sin användning av *programmering* i matematikundervisningen
- utmanas till reflektion över sina undervisningsbeslut, med särskilt fokus på *programmering*
- också tillägna sig en bredare uppsättning metoder och arbetssätt, med särskilt fokus på *programmering*

Innehåll - Modul II

- Om programmering
- Att undervisa med programmering
- Programmering med matematik
- Programmering i matematik

Orkestrering

Ett betydelsefullt begrepp i samband med undervisning i programmering är orkestrering.

Det handlar om det sätt på vilket samspelet mellan datorer, elever, läraren, tavlan och den gemensamma skärmen organiseras.

Orkestrering

Själva orkestreringen består med denna tolkning av både förberedelser och genomförande av undervisning:

- Förbereda:
  – didaktisk organisation (uppgifter, material, verktyg etc.)
  – plan för genomförande (interaktion, arbeta enskilt-par-alla, etc.)

- Genomföra:
  – implementera planen med stöd av den didaktiska organisationen.

Orkestrering

Exempel på frågor då lärare orkestrerar:

- Hur ser introduktionen av lektionen ut?
- Hur ska ni organisera? Verktyg? Elever? Arbetssätt?
- Vilka olika tillvägagångssätt tror ni eleverna använder?
- Hur planerar ni bäst för helklassdiskussionen i slutet av lektionen?

- Klassen arbetar tillsammans, med en gemensam skärm
  - Läraren förklarar matematik eller teknik
  - Läraren kopplar samman olika representationer av matematiska objekt
  - Läraren utgår från tidigare genomförda arbeten
  - En elev sköter teknologin

- Eleverna arbetar enskilt eller i par, läraren samlar eleverna vid en skärm
  - Läraren visar matematik och teknik under arbetets gång
  - Läraren ger spontana ”snabbgenomgång” när eleverna behöver det
- Eleverna arbetar enskilt eller i par, läraren går runt
  - Läraren lyssnar och samlar underlag för fortsatt undervisning

Undervisning i programmering

- Instruktions-orienterat
- Experiment-orienterat

Viktigt när man undervisar instruktionsorienterat

- Att veta vilka begrepp som är viktiga och att ha färdiga exempel på hur de används.
- Meningsfulla scenarier där de inlärda begreppen och färdigheterna kan användas.
- En lärogång där de mest grundläggande begreppen presenteras i en genomtänkt ordning.
- Det finns en risk att några elever tappar motivation och meningen med detta tillvägagångssätt, eftersom det blir programmering för programmeringens egen skull

Viktigt när man undervisar experimentorienterat

- Få till miljöer och scenarier där eleverna kan komma i gång med att formulera egna programmeringsprojekt
- Datorprogram som eleverna kan öppna/komma in i och ändra i ("tinkering")
- God förmåga att tala med eleverna om matematiken och datalogin i deras projekt
- Det finns en risk för att det blir svårt för vissa elever att klara undervisningsformen, för att ramarna blir för lösa

Undervisa i programmering

Två sätt skulle kunna vara:

- Genomgång av aktuella begrepp
- Tinkering

Tinkering

Komma underfund med hur något fungerar genom att mixtra med det.

I skolsammanhang:

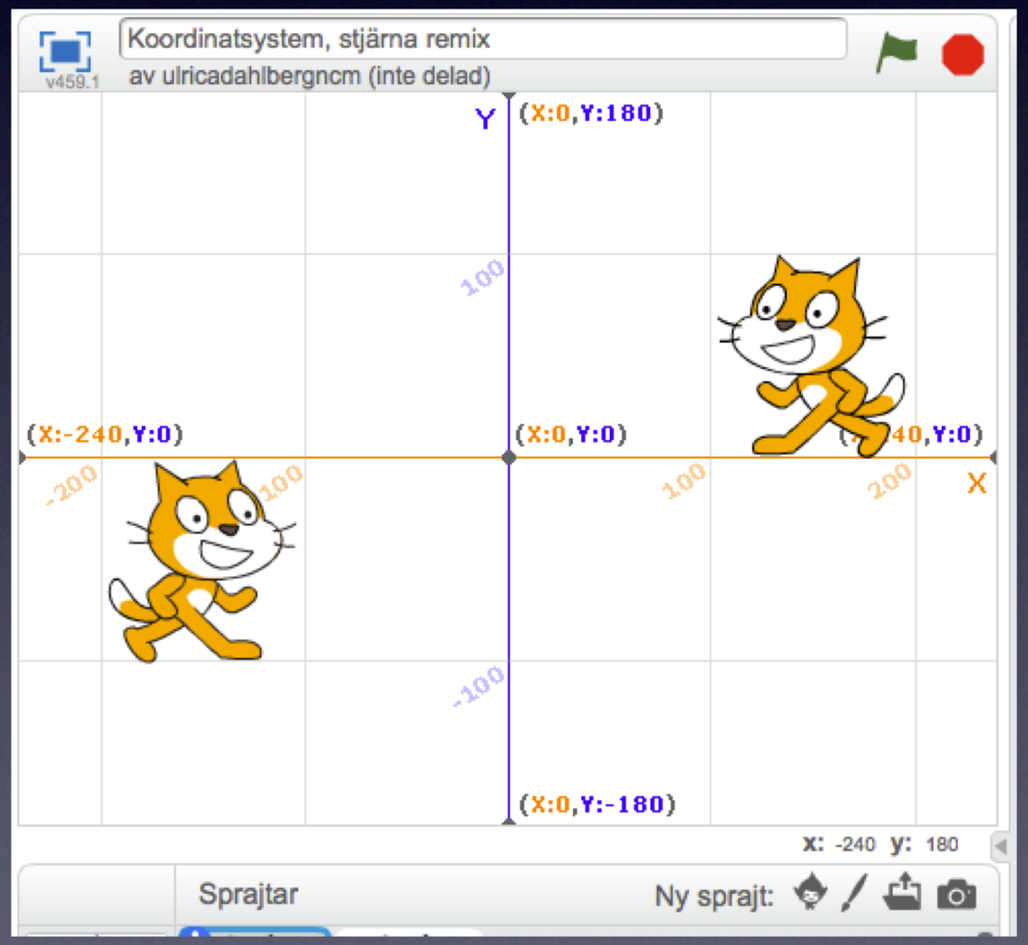
Elever kommer underfund med hur programmet fungerar genom att mixtra med det.

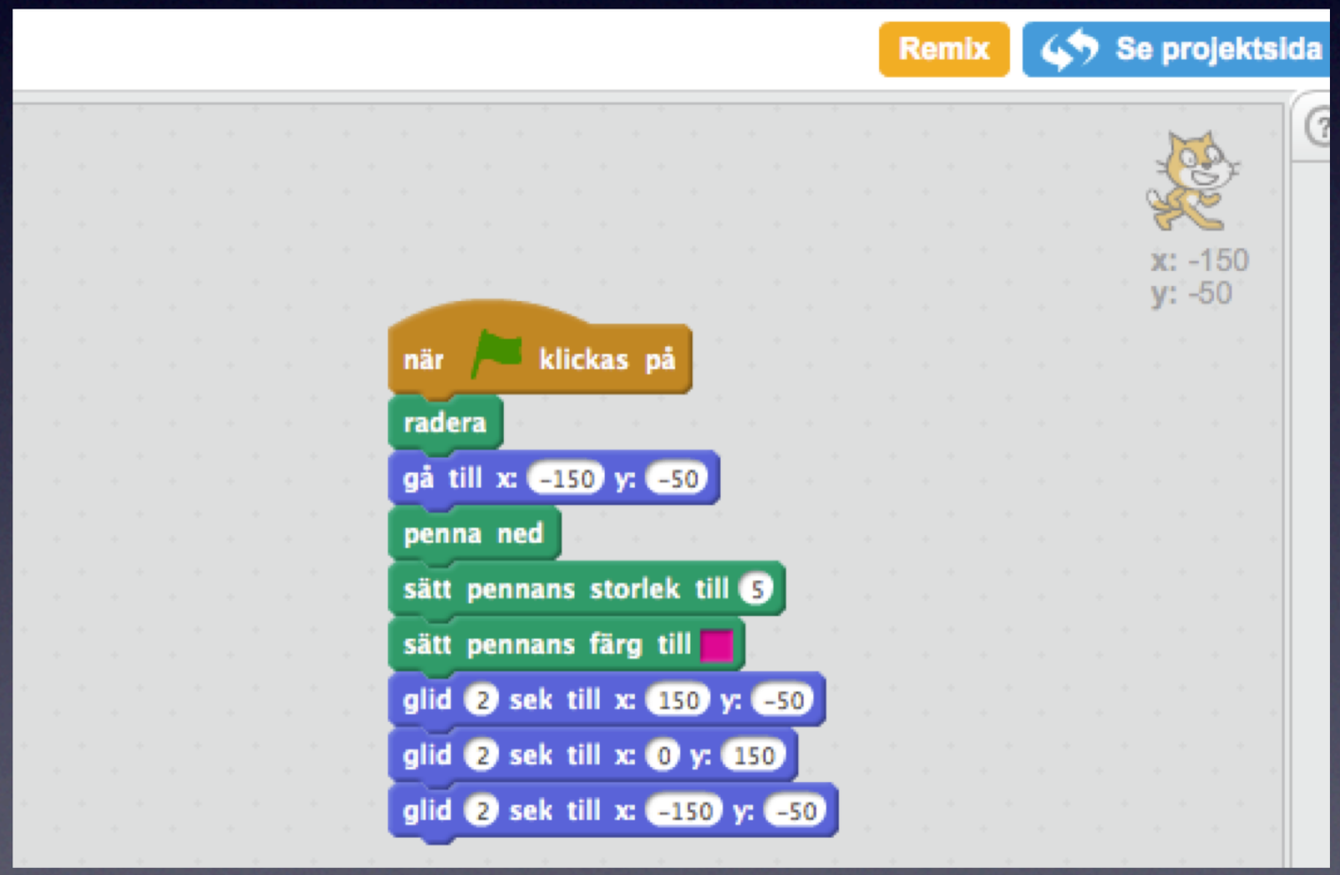
Orkestrering

- Förutse
- Överblicka
- Välja ut
- Ordna
- Koppla ihop

Aktivitet med Tinkering

- Öppna Katten Scratch ritar en stjärna





Vad är programmering?

Programmeringens olika faser

- Problemanalys
- Sekvensera problemet logiskt
- Skriva ett datorprogram, koda
- Exekvering
- Felsökning (alt ord avlusning eller debugging)

Testa två och två

Programmera med ”pseudokod”:

Skriv ett program för att göra en marmeladsmörgås.

Att programmera

Följa en sekvens

Programmering i matematikundervisningen: två sidor

1. Programmera något som i sig inte är matematiskt
2. Programmera matematik

NCM Nationellt centrum för matematikutbildning

GÖTEBORGS UNIVERSITET

Programmering i matematikundervisningen: två sidor

1. Programmera något som i sig inte är matematiskt
2. Programmera matematik

NCM Nationellt centrum för matematikutbildning

GÖTEBORGS UNIVERSITET

1. Programmera något som i sig inte är matematiskt

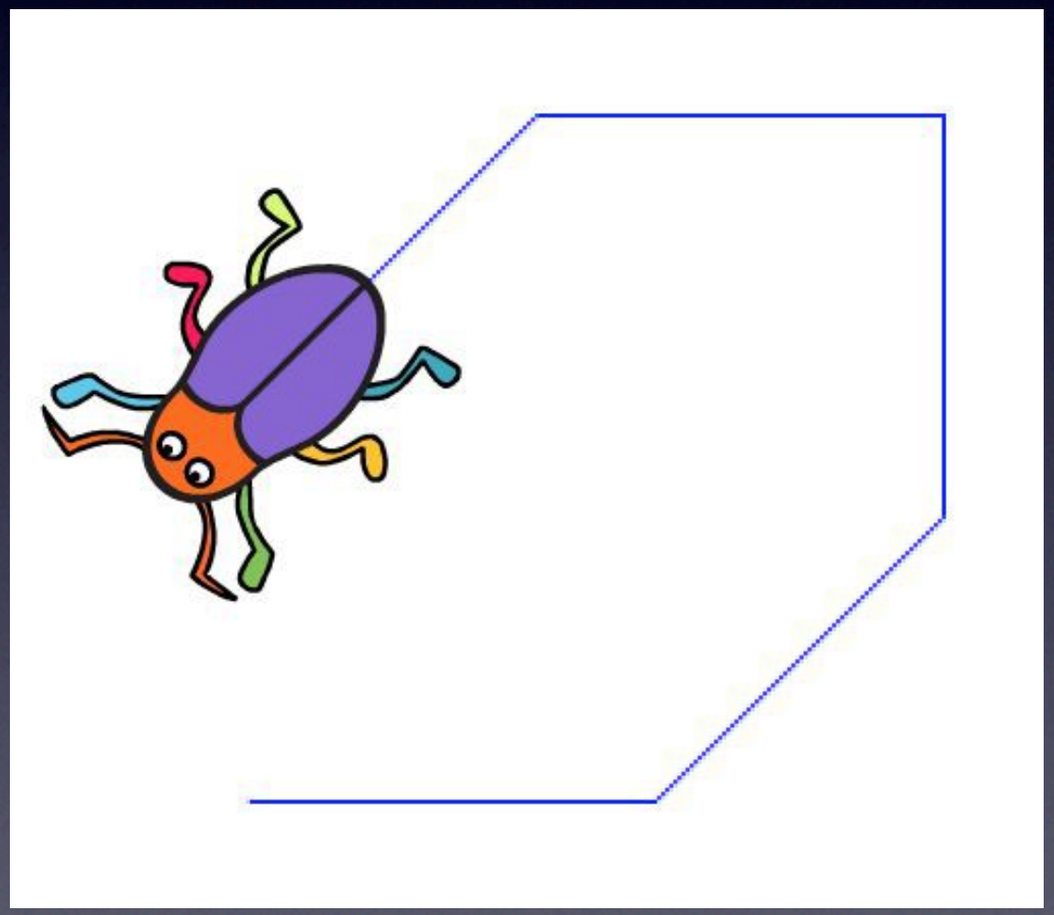
Hur kan matematik och programmering stärka varandra?

- Algoritmer och matematik
- Matematiken i kod och modeller
- Matematiskt tänkande och kunnande kan utvecklas genom programmering

Representation av samma situation på olika sätt

Kod

Grafisk representation



Att undervisa för att stödja samspelet mellan matematik och programmering

- Utforma programmeringsuppgifter
- Låta eleven förklara det matematiska innehållet
- Läraren identifierar matematiken i elevernas aktiviteter
- Snabbgenomgångar

2. Programmera matematik

A. Lös ett matematisk problem

B. Skapa ett matematiskt verktyg

C. Undersök en matematisk idé

Aktivitet

- Sållmetoden – hitta primtalen upp till 100.
- Hur skulle en *pseudokod* se ut för att finna ett primtal?
- Börja att skriva denna kod i ett programspråk (Scratch, Python, Javascript …)

Eratosthenes sållmetod

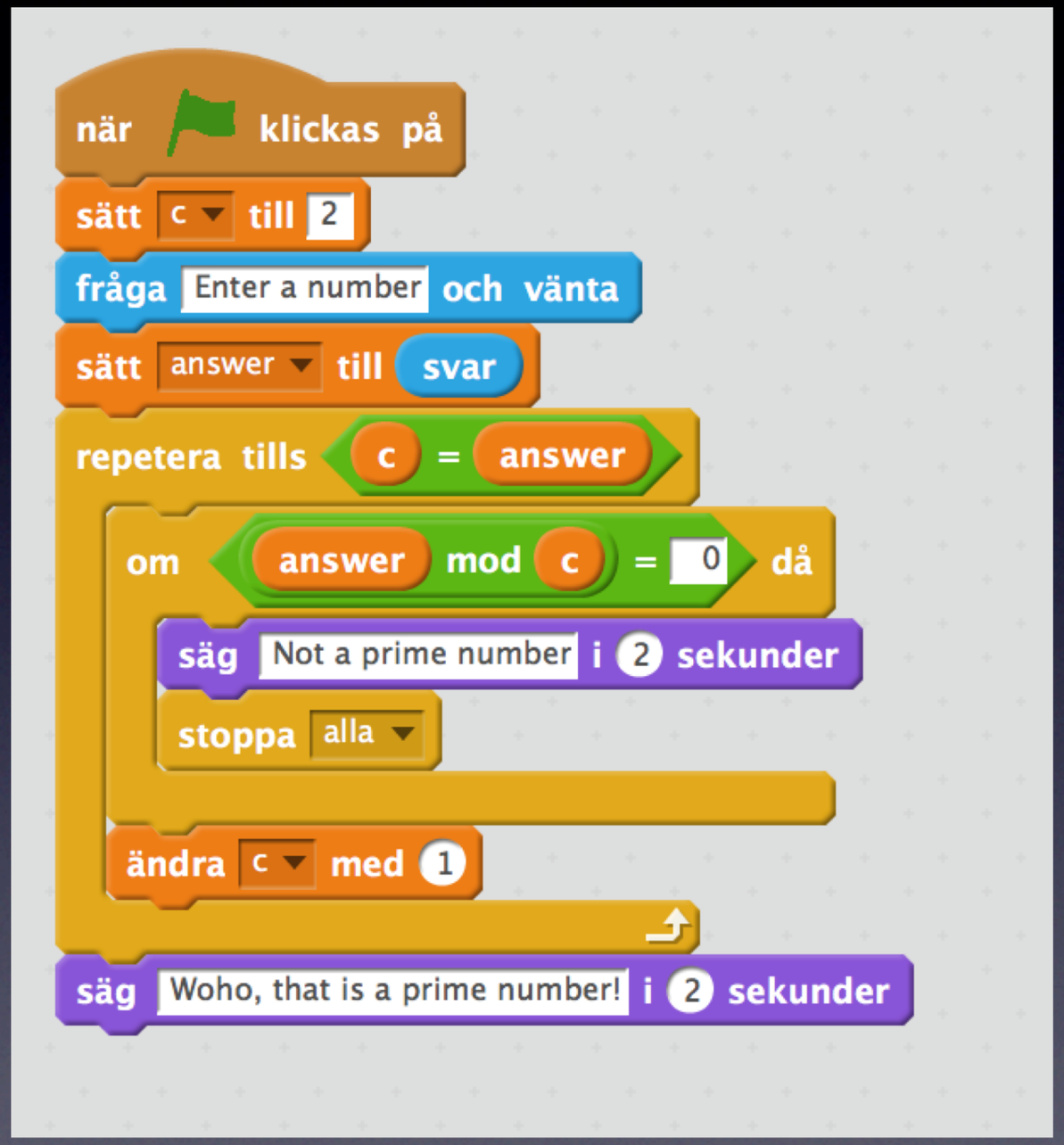
1. Gör en lista över alla tal från 2 till 100.
2. Stryk i listan över alla jämna tal, dvs delbara med 2, som är större än 2 (4, 6, 8 osv.).
3. Nästa tal på listan som inte är överstruket är ett primtal.
4. Stryk över alla tal, som både är större än det primtalet du hittade i föregående steg och multiplar av det.
5. Upprepa stegen 3 och 4.
   Alla kvarstående tal i listan är primtal.

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

Möjlig pseudokod

```
Sätt nämnare = 2

Sätt tal = [siffran du vill undersöka]

Medan nämnare är mindre än tal

    Om tal/nämnare=jämt Skriv ”Tal är INTE primtal”

Annars

    öka nämnare med 1
```

Scratch


när klickas på
sätt c till 2
fråga Enter a number och vänta
sätt answer till svar
repetera tills c = answer
om answer mod c = 0 då
säg Not a prime number i 2 sekunder
stoppa alla
ändra c med 1
säg Woho, that is a prime number! i 2 sekunder

Python

Python

```
tal = int(input("Ditt tal:"))

for a in range(2,tal):

  if (tal%a == 0):

    print(tal, ",INTE primtal")

    break

else:

  print(tal, "är ett primtal")
```

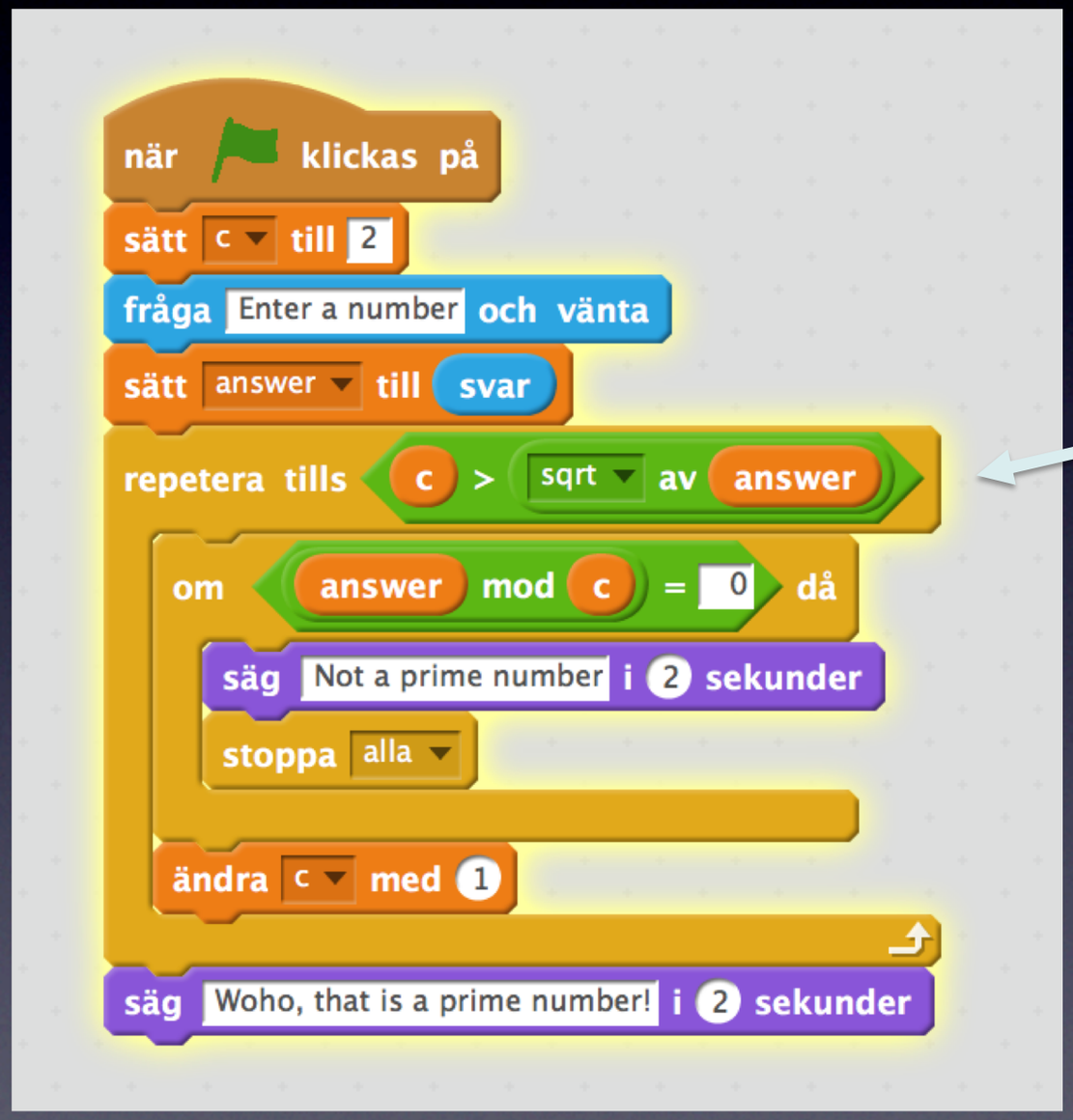
Kör programmet

Javascript

Javascript

```
check=true;
tal = int(input("Ditt tal:"))
for (a = 2; a < tal; a = a +
1){
   if(tal%a == 0){
     print(tal, ",INTE
primtal")

     check = false;

     break;
   }
}
if(check)
   print(tal + " är ett
primtal!");
```

Ändrad kod i Scratch



Diskutera

Vilka matematiska problem kan vi nu knyta till det här verktyget vi just byggt? När vore det lämpligt att ha ett verktyg som just bestämmer primtal eller undersöker om ett tal är primt?

Problem med primtal

Jacob skrev ner fyra konsekutiva (på varandra följande) positiva heltal. Han beräknade sedan de fyra möjliga summor man får då tre av talen adderas. Ingen av dessa summor var ett primtal. Vilket är det minsta tal som Jacob kunde ha skrivit?

Problem med primtal

- “Om n är ett primtal så är exakt ett av talen n–2 och n+2 ett primtal”
- Stämmer detta påstående? För alla primtal?
- Om inte, när gäller det inte?

Delbarhet

- Vilken rest får vi om vi delar tal med 12?
- Vad kan man förvänta sig för rester när man delar ett tal med ett annat?

Diskussionsfrågor

- Beskriv för varandra vilka erfarenheter ni har i och om programmering. Hur kan vi ta hjälp av varandra och utnyttja varandra?
- Hur ser er matematikundervisning ut utan programmering? Vad är det som avgör när man väljer instruktionsorienterat eller experimentorienterat? När väljer man det ena eller det andra?
- Vilka konkreta företeelser (tex rörelse, position, antal, form, ränta …) eller matematiska begrepp kan du koppla till den programmeringsuppgift ni just prövat?