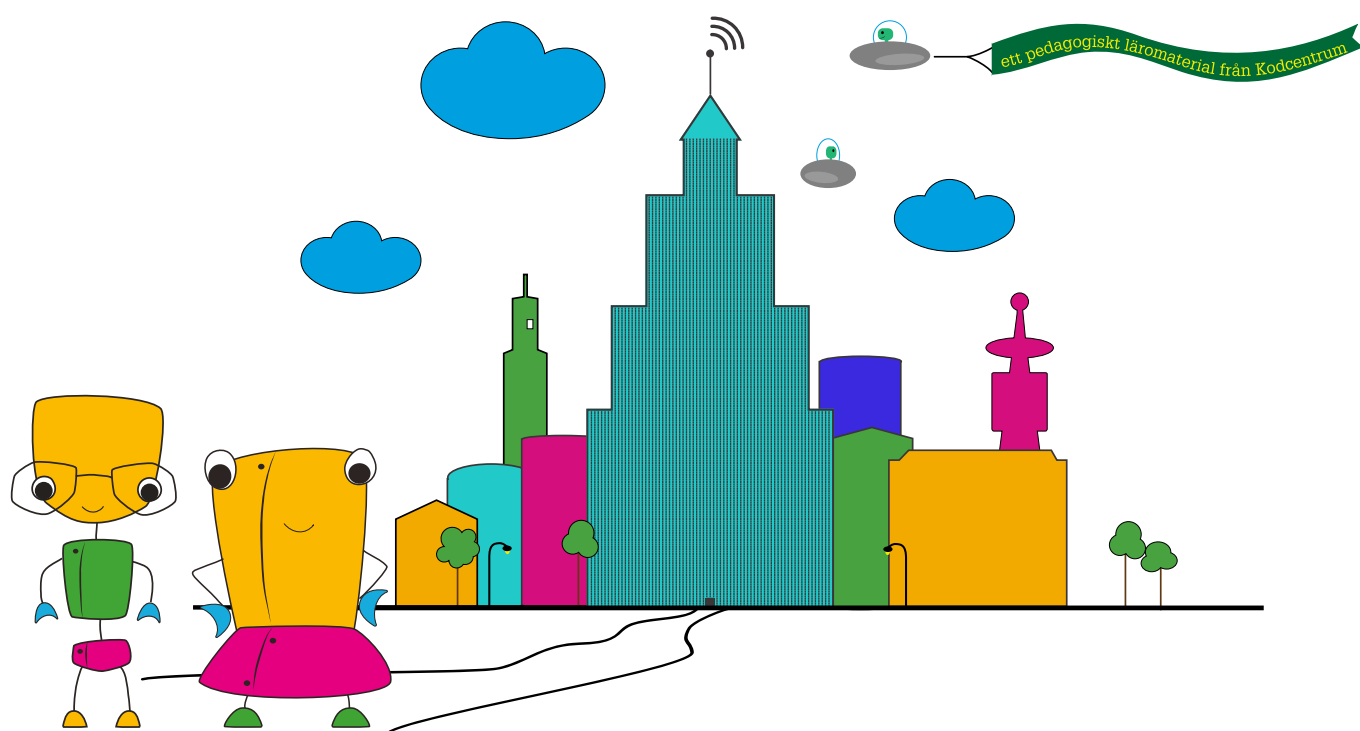


ROBOTDETEKTIVVERNA

10 datalogiska utmaningar i Nordopolis



Hej!

Det här är berättelsen om robotarna Else och Random som bor i Nordopolis, en framtida stad där robotar och människor lever tillsammans. Berättelsen innehåller 10 datalogiska utmaningar som på ett lekfullt och klurigt vis övar dig i datalogiskt tänkande - hur du kan titta på problem på ett sätt som liknar hur en dator skulle lösa dem.

Datalogiskt tänkande är en viktig del av att programmera. Det handlar bland annat om att bryta ned problem i mindre delar, hitta mönster och söka efter felaktigheter, buggar. Det handlar om att kunna resonera sig fram och förklara för någon annan hur du har tänkt dig lösa problemet. Det innebär också att vara kreativ och fundera ut möjliga lösningar eftersom det ofta finns flera olika sätt att lösa problem på!

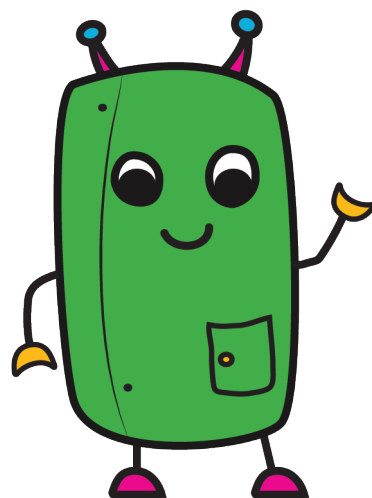
Utmaningarna i berättelsen är på tre nivåer, markerade med 1-3 pusselbitar:

Nivå 1: Det är ingen kod i dessa uppgifter, men det är bra om du har fått en introduktion till programmering innan.

Nivå 2: Dessa innehåller kodblock. Du behöver ha programmerat lite tidigare och känna till vad repetition, villkor och variabler innebär.

Nivå 3: Här är uppgifterna lite klurigare, koden är längre och skriven i så kallad pseudokod. Du bör vara bekväm med repetition, villkor och listor.

Ett par av utmaningarna är inspirerade av Bebras, en tävling i datalogiskt tänkande för barn och unga i åk 2-9 samt gymnasiet. Läs mer på bebras.se.



Ett pedagogiskt läromaterial
från Kodcentrum.

Text: Marie Gustafsson Friberger
Illustrationer och formgivning:
Elinor Hägg och Erika Granstrand

www.kodcentrum.se
www.kodboken.se

<Kod
centrum>

Innehåll:

<Start>

Utmaningar nivå 1

- 1 – Någon har stulit mitt minneskort!
- 2 – Lägg mojängen i verktygsfacket
- 3 – Hemliga meddelanden

Utmaningar nivå 2

- 4 – På patrull
- 5 – Loopa ut från lagret
- 6 – Varför får inte pappa samma reklam som jag?

Utmaningar nivå 3

- 7 – Kod röd på våning 13
- 8 – Debugga värmen
- 9 – Samla på spår
- 10 – Var är robotkatten?

</Slut>

<Start>

- Vi kom in, vi kom in! Random springer fram till Else, hoppar upp och ner och ler stort.
- Vad roligt, säger Else och småler, men verkar inte lika glad som Random. Else har nämligen inte programmerats med lika utvecklade känslouttryck som Random.
- Oj, det står i meddelandet att vi börjar redan imorgon kl 9, läser Random. Vi ska infinna oss på våning 101.
- Dit har vi aldrig fått tillgång innan, säger Else med viss förtjusning. Jag undrar vad vår första utmaning blir.

Året är 2218 och platsen är Nordopolis. Nordopolis är en tornstad en bit norr om Väneren där 200.000 människor och robotar bor. Nordopolis består av ett jättehögt och ganska brett hus med 200 våningar och åtta mindre stödbyggnader runtomkring.

Robotarna Random och Else är två av åtta robotar som kommit in på detektivskolan. De har valts ut i hård konkurrens bland många olika sorters robotar. Random har drömt om att bli detektiv i flera år, älskar att lösa kluriga mysterier och vill förstå koden bakom Nordopolis. Else tycker att Nordopolis är lite tråkigt och hoppas att det blir mer äventyr som detektiv.





Utmaning 1:

Någon har stulit mitt minneskort!

Alla robotdetektiver ingår i olika team som har olika uppdrag. För att snabbt kommunicera viktiga meddelanden till ett team används ett antal hemliga symboler som visas på robotarnas informationsskärmar. Varje meddelande består av tre symboler.

Team-symbolen visar vilket team som meddelas:

- Random tillhör Team 1
- Else tillhör team 2

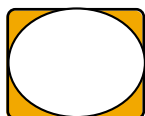
Orsaks-symbolen visar vad som har hänt:

- Stöld
- Dataintrång
- Drönarolycka

Tids-symbolen visar hur snabbt teamet behöver samlas:

- Bråttom – kom genast till teamrummet
- Inte bråttom – kom till teamrummet om en timme

Team-symboler

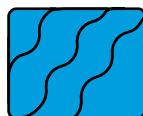


Team 1



Team 2

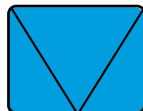
Orsaks-symboler



Stöld



Dataintrång

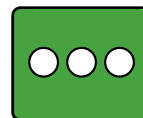


Drönarolycka

Tids-symboler



Bråttom

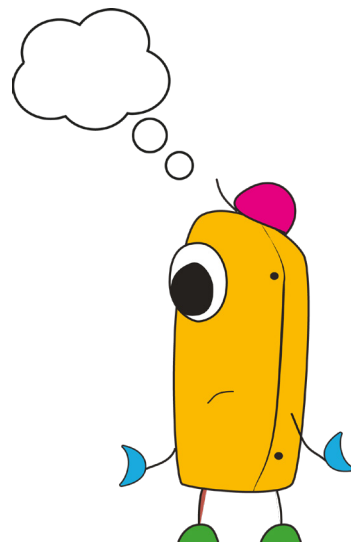


Inte bråttom

Frågor:



- 1. Vad betyder meddelandet här?
- 2. Någon har stulit robotens Repeats minneskort. Det är bråttom. Skicka ett meddelande till Else:s team!
- 3. Kom på en egen symbol. Vad kan den betyda?
- 4. Gör ett meddelande där den nya symbolen ingår och låt din kompis tyda det.
- 5. Spelar ordningen på symbolerna någon roll eller inte, för denna utmaning?



Vad har detta med datorer och kod att göra?

Denna utmaning har att göra med datorer och kod på två sätt: den handlar om symboler och deras ordning.

Inom datavetenskap använder man symboler, det vill säga något som kan tolkas om till något annat, på flera olika sätt. Ett exempel på hur symboler används är hur siffror och bokstäver i stället kan skrivas med binärkod – ettor och nollor. Ett annat exempel är hur vissa tecken används för att betyda specifika saker i programmeringsspråk. Kanske har du sett kod med symbolerna { och } till exempel?

Inom programmering är det ofta viktigt i vilken ordning (kallas även för sekvens) som kommandon utförs. Har du programmerat ett spel, så vet du att det är viktigt att man får poäng och förlorar liv vid rätt ställe. Det är däremot inte alltid som ordningen är viktig, som i utmaningen ovan, och det



Utmaning 2:

Lägg mojängen i verktygsfacket

Nu är det dags för robotarna att få sina detektivverktyg och utföra övningsuppdrag för att lära känna verktygen. De verktyg som en ny robotdetektiv behöver är en mojäng, detektor och scanner.

Verktygen förvaras i en inbyggd verktygsförvaring i robotens "mage". I verktygsförvaringen finns bara plats för sex verktyg och den är uppdelad i två fack. Roboten kan bara komma åt det yttersta verktyget i vardera fack. I ena facket läggs de verktyg som ska användas på uppdraget. När de sedan har använts läggs de i det andra facket.



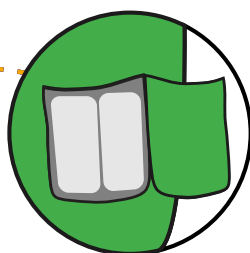
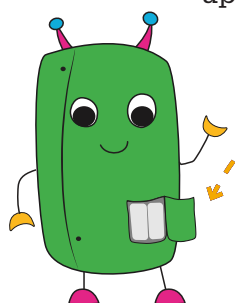
Mojäng



Detektor



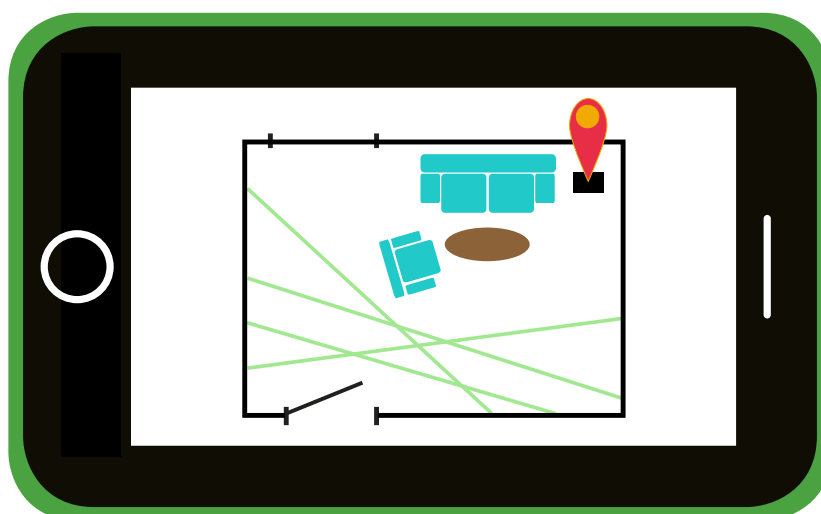
Scanner



Övningsuppdragen sker på tid och det är viktigt att verktygen ligger i rätt ordning: det som ska användas först ska ligga längst ut och det som ska användas sist ska ligga längst in.

Så här går övningsuppdraget till:

Först ska robotarna kolla så att det inte finns några osynliga fällor, då används scannern. Sedan ska den hemliga lådan i rummet hittas med hjälp av detektorn. När de hittat lådan öppnas den med hjälp av mojängen. Så här ser uppdraget ut för robotteamet som övervakar övningen:



Frågor:

- 1. I vilken ordning ska verktygen ligga i verktygsfacket för att robotarna ska kunna utföra uppdragen optimalt (så bra och snabbt som möjligt)?
- 2. Skriv ett nytt uppdrag till robotarna där detektorn används först och scannern sist.
- 3. Hur skulle verktygsfacket i roboten kunna designas för att det verktyg som stoppas in först också kan komma åt först?

Vad har detta med datorer och kod att göra?

Exemplet med verktygsförvaringen kan liknas vid en datastruktur. I programmering används datastrukturer för att ordna data (t ex ord eller siffror) så att datorn effektivt kan lagra och komma åt dem. Det finns olika sorters datastrukturer, där en *lista* är den enklaste formen.

Facken i robotarnas verktygsförvaring är som datastrukturen *stack* (engelskans stapel), där det man stoppar in sist tas ut först. Det är lite som en tallriksstapel i skolmatsalen – det går bara att ta den tallrik som ligger överst, som är den som sist lagts på stapeln. Motsatsen, att det man stoppar in först tas ut först, kallas för en *kö*.

Använder du ett musikprogram där du kan lägga till låtar i en lista för att spelas lite senare? Lägg de i en *kö* eller *stack*? Vad är bra med det sättet? Vad är sämre?



Utmaning 3:

Hemliga meddelanden

Detektivrobotarna skickar elektroniska meddelandet till varandra. Ifall någon bov skulle hacka deras system och få tag på meddelandena är det viktigt att använda kryptering för att dölja meddelandets innehåll.

Så här fungerar deras kryptering. Om Else ska skicka meddelandet JAGHARFÅNGATCHOKLADTJUV använder den ett program som skriver meddelandet i en tabell med fem kolumner. Meddelandets första bokstav skrivs i den översta kolumnen till vänster i tabellen och sedan följer bokstäverna från vänster till höger, rad för rad. När meddelandet är slut fyller programmet på den sista raden med A:n.

Så här blir det:

J	A	G	H	A
R	F	Å	N	G
A	T	C	H	O
K	L	A	D	T
J	U	V	A	A

För att skapa det hemliga meddelandet skriver Else sedan varje kolumn uppifrån och ned, från vänster till höger. Det hemliga meddelandet blir: JRAKJAFTLUGÅCAVHNHDAAGOTA

Frågor:

- 1. Else får svaret ÄELTULIPTNPATAA. Vad betyder det?
- 2. Vilken information är viktigast att känna till för att kunna avkoda ett meddelande med denna typ av kryptering?
- 3. Skicka ett meddelande till en klasskompis med denna typ av kryptering. Använd fyra eller fem tabellkolumner!
- 4. Kolla på meddelandet från din klasskompis. Kan du lista ut om det är fyra eller fem tabellkolumner? Hur gör du det? Vad står det i meddelandet?
- 5. Hur bra är denna sorts kryptering på att dölja information, tror du?

Vad har detta med datorer och kod att göra?

Om du vill skicka meddelanden som bara mottagaren ska förstå behöver meddelandet krypteras. För att det ska fungera behöver den som tar emot meddelandet veta hur den ska kunna avkoda (förstå) meddelandet. Den behöver kunna känna till *nyckeln*. I det här fallet är nyckeln att veta hur många tabellkolumner som ska användas.

Kryptering som bygger på att du flyttar bokstäverna, som du gör i uppdraget ovan, kallas för transpositionsskiffer. Ett känt exempel på detta skiffer kommer från antikens Grekland och kallas *Scytale* efter den cylinder som användes. På den rullades ett pergament (papper) för att göra förflyttningen av bokstäver. För att det skulle fungera behövde både den som skickade och tog emot meddelandet ha samma storlek på cylindern.

Skiffret i exemplet är ganska dåligt, eftersom det är lätt för någon att komma på hur stor tabell som används. I modern kryptering används mer komplicerade sätt att dölja innehållet i meddelanden. Ett exempel är HTTPS (Hyper Text Transfer Protocol Secure) där kryptering används för att obehöriga inte ska kunna läsa av datatrafik som skickas från din webbläsare till webbsajter som du använder.

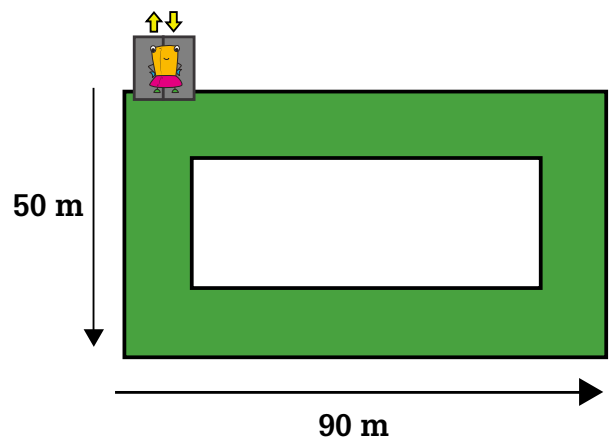
Utmaning 4:

På patrull

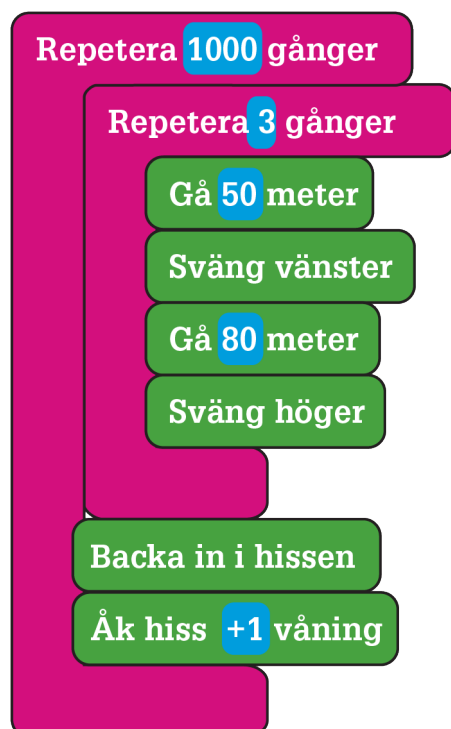
Else och Random är klara med detektivskolans första nivå. Nu får de börja komma ut på riktiga uppdrag, men de har fortfarande en del kvar att lära innan de får sin detektivlicens.

En gång om dagen patrullerar en robotdetektiv igenom alla 200 våningar i Nordopolis. Den letar efter spår till några av de vanligaste typerna av brott i Nordopolis. Idag är det Elses tur att patrullera.

Huvudkorridoren går ett helt varv inuti Nordopolis, en huvudkorridor på varje våningsplan. Bilden visar ungefär hur huvudkorridoren ser ut. Det finns hissar i varje hörn, men bilden visar bara den hiss som Else använder just nu.



Blocken nedan visar den kod som körs för att ta sig runt vid patrulleringen. Samtidigt körs en annan kod för att leta spår. Det har blivit fyra fel, "buggar", i koden som används för att patrullera. Kan du hitta dem?



Frågor:

- 1. Kan du hitta de fyra buggarna i koden? Hur ser koden ut när den är fixad?
- 2. När koden är fixad, hur många gånger kommer kommandot "Gå 50 meter" att utföras?
- 3. Skapa en ny bugg (ett fel) i koden. Visa för en kompis. Kan den hitta buggen?

Vad har detta med datorer och kod att göra?

När det blir fel i kod brukar man kalla det för en *bugg*. Det kommer från engelskans bug, det vill säga insekt. Ordet har faktiskt använts för fel i maskiner redan innan vi hade datorer och programkod, det användes bland annat av den amerikanske uppfinnaren Thomas Edison på 1800-talet. Det är oklart om ordet kommer sig från insekter som krupit in i maskinerna, eller för att ordet också kan betyda monster!

För att hitta buggar i ett program kan du köra programmet "i huvudet" och fundera på vad som händer i varje steg. Här kan det krävas både fantasi och vana, du blir bättre ju mer du övar! Kör du istället ett program för att hitta fel gäller det att vara uppmärksam på hur det du ser hänger ihop med koden: I vilket steg i koden är det som det inte blir som du tänkt dig?

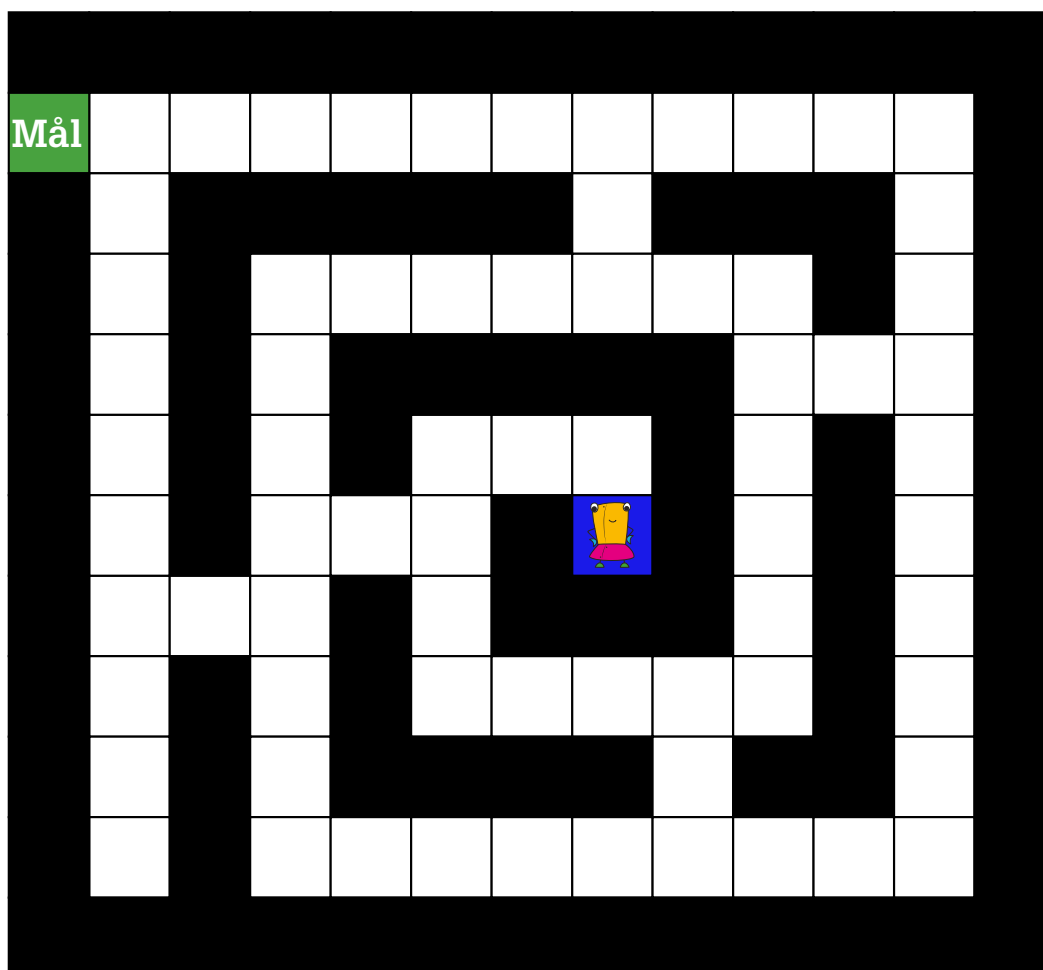


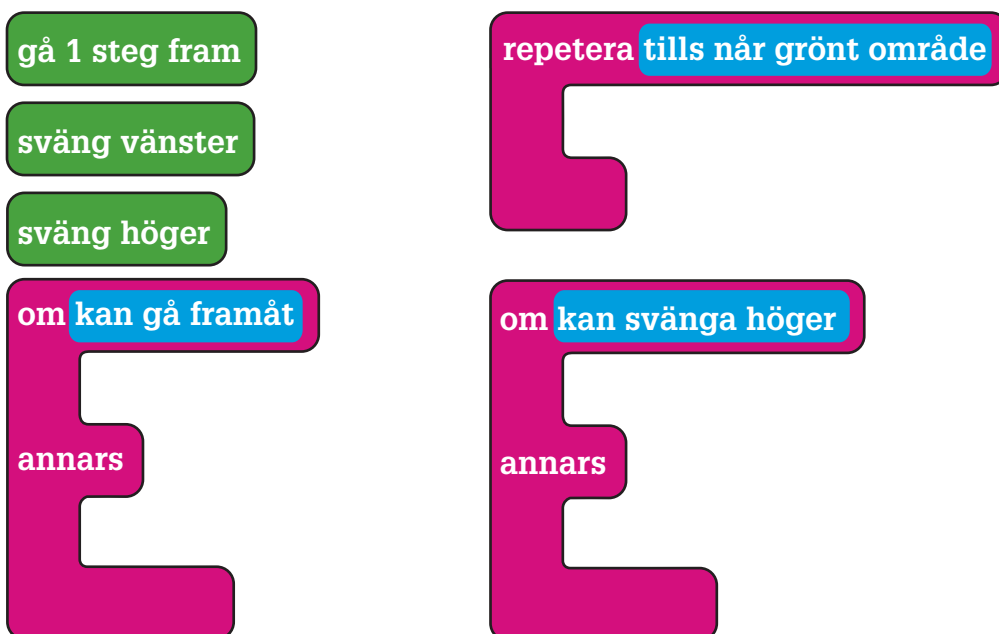
Utmaning 5:

Loopa ut från lagret

Å nej! Else har hamnat i knipa och måste ta sig ut ur en lagerbyggnad. Men det är rena labyrinten! Dessutom har ett av Elses minneskort gått sönder och kan därför bara köra korta program.

Else står vid det blåa fältet och måste ta sig till det gröna fältet, med ett program som bara använder fyra kodblock. På nästa sida ser du de kodblock som du kan välja på. Kan du klura ut vilka fyra och i vilken ordning för att Else ska komma ur knipan? De vita blocken är korridorer och de svarta blocken är väggar.





Frågor:

- 1. Gör ett program (rita på papper) som använder endast fyra kodblock för att komma från det blåa startfältet till gröna målfältet.
- 2. Kan du rita en egen labyrint som fungerar att ta sig från start till mål med bara fyra kodblock? Låt en kompis klura ut hur programmet ska se ut!

Vad har detta med datorer och kod att göra?

För att lösa uppgiften behöver du undersöka de möjliga vägarna att gå från startpunkten till slutpunkten. Du behöver sedan se om någon av dessa vägar följer ett mönster som går att beskriva med ett fåtal kodblock. Detta är exempel på datalogiskt tänkande!

Kodblocken som används illustrerar viktiga delar i programmering: *villkor* (om-då-annars) och *repetition*, det som också kallas för loopar. Här repeterar du koden tills något nytt händer i programmet.

Utmaning 6:

Varför får inte pappa samma reklam som jag?

De flesta som bor i Nordopolis använder sociala nätverk för att hålla kontakt med familj och vänner från alla delar av världen. Många av dessa nätverk är gratis, men den som använder dem får se reklam.

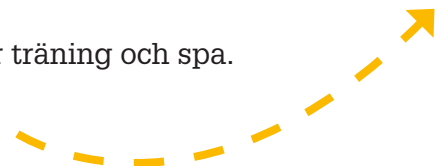
Robotdetektiverna har fått besök. Det är Felicia som undrar varför hon och hennes pappa får olika reklam på det sociala nätverket Månstorm. Random och Else får i uppdrag att undersöka saken.

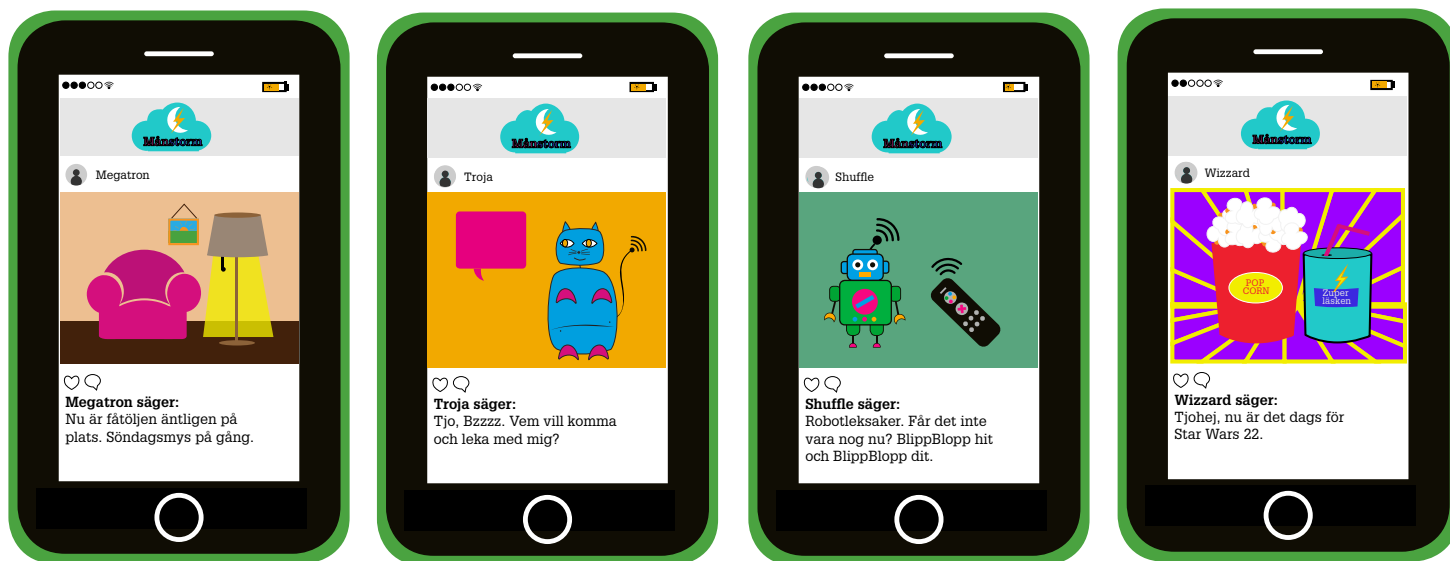
Det visar sig att Månstorm bland annat använder ålder för att bestämma vilken reklam som visas. Men användaren matar inte in sin ålder. I stället analyserar Månstorm åldrar genom att poängsätta användarens texter och bilder med hjälp av en algoritm som har ett antal regler:

- Om det är bilder på coola robotleksaker antar de att det är ett barn och inlägget får +3 poäng.
- Om det är bilder på heminredning antar de att det är en vuxen och inlägget får -3 poäng.
- Om meddelandet innehåller något av orden "Tjohej", "Bzzzz" eller "BlipBlop", ord som är populära bland barn vid tillfället, så får inlägget +2 poäng.
- Om meddelandet innehåller långa ord (10 bokstäver eller mer) är det nog en vuxen, så inlägget får -2 poäng.

För att avgöra vilken reklam som ska visas tillsammans med ett inlägg räknas poängen ihop. Vad som visas beror på summan:

- Om summan är +2 eller mer visas reklam för nya robotleksaker.
- Om summan är mellan -1 och +1 visas reklam för senaste hologramfilmen, en familjefilm för både vuxna och barn.
- Om summan är -2 eller mindre visas reklam för träning och spa.





Inlägg 1

Inlägg 2

Inlägg 3

Inlägg 4

Frågor:

- 1. Vilken reklam kommer att visas för de fyra inläggen som du ser här?
- 2. Ett av inläggen är skrivet av en vuxen, men algoritmen tolkar det som att det är ett barn. Kan du gissa vilket? Varför fungerade inte algoritmen för detta inlägg?
- 3. Kan du skriva ett nytt inlägg där reklamen som kommer att visas blir för den senaste hologramfilmen?

Vad har detta med datorer och kod att göra?

Denna typ av poängsättning som används i utmaningen används i flera sammanhang, även om de algoritmerna ofta är mycket mer avancerade. Förutom att avgöra vad som ska visas i sociala nätverk används liknande beräkningar för att bestämma i vilken ordning som sökresultat ska visas i sökmotorer.

För att bli medveten om hur information om dig används på nätet kan det vara bra att känna till dessa algoritmer. Om du använder sociala nätverk eller sökmotorer kan du ofta se vilka intressen de har räknat fram att du har, utifrån vilken reklam som visas. Testa om du kan påverka reklamen genom att skriva om, eller söka på, något särskilt!



Utmaning 7:

Kod röd på våning 13

Nu är Random och Else färdiga med detektivskolans andra nivå och de har inte långt kvar förrän de får sina licenser. Men hur ska det gå, kommer de att klara Nordopolis lite klurigare uppdrag?

Brrring! Det kommer in ett larm från ett lager på våning 13. Någon måste dit och kolla vad som har hänt.

När larmet kommer anropas den robot som är närmast. I vissa fall är det farligt att undersöka själv, till exempel om det kan vara en inkräktare kvar i lokalen eller om det är risk för brand. Då är det "kod röd". På våning 100-102 kryllar det av detektiver och det går att få hjälp jättefort, så där kan roboten alltid undersöka själv.

För att robotarna Else och Random ska få rycka ut på larm programmeras de med nya regler. Vad de ska göra beror på vilken våning larmet är från och om det är "kod röd":

Om våning < 100 då

Om kod röd då

Anropa förstärkning

Annars

Undersök närmare

Om våning >=100 och våning <= 102 då

Undersök närmare

Om våning >= 103

Om kod röd då

Anropa förstärkning

Annars

Undersök närmare



Frågor:

- 1. Vad händer om larmet kommer från våning 13 och det är kod röd?
- 2. Vad händer om larmet kommer från våning 100 och det är kod röd?
- 3. Robotdetektiverna bestämmer sig för att förenkla sina regler. Om det är "kod röd" så ska förstärkning alltid anropas, oavsett vilket våningsplan de är på. Hur ser koden ut då?

Vad har detta med datorer och kod att göra?

Villkor används i programmering för att det ska kunna hända olika saker när ett program körs beroende på vad den som använder programmet gör. Om du programmerar spel i Scratch kanske du har använt *villkor* för att kolla om två sprajtar rör vid varandra om en sprajt är nära en viss koordinat.

Villkor används alltså för att programmet ska kunna välja mellan olika alternativ (om-då-annars). I denna utmaning är det extra tydligt eftersom ett beslutsträd formas av villkoren. Det kallas beslutsträd för att de olika alternativen formar grenar (det blir tydligare om du vänder på pappret).

Du kan lägga villkor i villkor som i exemplet ovan. Men det gäller att hålla tungan rätt i mun, ibland blir det svårt att hålla redan på vad som händer! Då är det bra att ha tänkt igenom koden innan du börjar programmera, så att du vet vilket alternativ som väljs när. Ofta kan du skriva villkor i villkor på olika sätt: Ska robotdetektiverna börja med att kolla våning eller kod röd – vilket blir lättast att förstå?



Utmaning 8:

Debugga värmen

Någon har hackat värmesystemet och ändrat i koden! Fastän det bara är 12 grader Celcius i Nordopolis är värmen inte på. Varför slår värmen inte igång? Ingen har lyckats lösa problemet och robotdetektivena får rycka ut. De misstänker att en hackare har varit i farten!

Här är koden som styr värmeläget. I den har hackaren ändrat ett tecken:

Temperatursensor = F

Om temperatur > 19

Sätt värmeläge till 0

Om temperatur > 12 och temperatur <= 19

Sätt värmeläge till 1

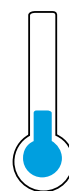
Om temperatur > 7 och temperatur <=12

Sätt värmeläge till 2

Om temperatur <= 7

Sätt värmeläge till 3

Robotarna kollar vad värmesystemet tror att temperaturen är: 54 grader Fahrenheit! Vad kan ha hänt?



Frågor:

- 1. Vad är det hackaren har ändrat i koden?
- 2. Hur skulle problemet, buggen, kunna lösas?

Vad har detta med datorer och kod att göra?

Fel i programkod kan ta många olika former. Att felsöka ett program som inte fungerar kan vara frustrerande, men när du hittar felet blir du oftast desto gladare.

När du felsöker är det viktigt att du håller dig lugn, funderar på vad orsakerna till problemet kan vara och hur du kan testa dem. Det är också bra att anteckna vad du redan har testat. Ibland blir det lättare att hitta felet när du tar en paus eller om du ber om hjälp - för då måste du förklara vad du gjort för någon annan. Kanske kommer du då på felet själv!

Ibland är det svårt att testa program på förhand och konsekvenserna av fel kan bli väldigt stora. Ett exempel på det är Mars Climate Orbiter, en farkost som skickades upp för att ligga i omloppsbana runt Mars. Tyvärr krashade den på grund av att två team med programmerare hade använt olika enheter utan att de visste om det.



Utmaning 9:

Samla på spår

Varje dag patrullerar robotdetektiverna genom Nordopolis, hela vägen från våning 1 till 200. Vad letar de då efter? Olika typer av spår av aktiviteter som inte är välkomna på Nordopolis. Hittar de smulor kan den vara en kaktjuv på spåren. Slime på väggarna kan innebära en alieninvasion. Och krossade minneskort – en robotmisshandel!

Vid varje genomgång av Nordopolis skapar roboten en *lista* av alla spår som hittats. En lista kan ses som en låda med många fack där du kan lägga information. Du kan säga vilket fack det ska läggas i, men om något läggs till utan att du säger var, läggs det bara på nästa tomma plats.

Idag är det Random som ansvarar för listan. Listan är tom när patrulleringen börjar. När ett spår hittas läggs det till i listan, tillsammans med en variabel för vilken våning det hittades på. När Random patrullerar från våning 1 till 200 körs följande kod:

Repetera (för alltid)

Om (smulor)

Lägg till i lista (våning + " Smulor")

Om (slime)

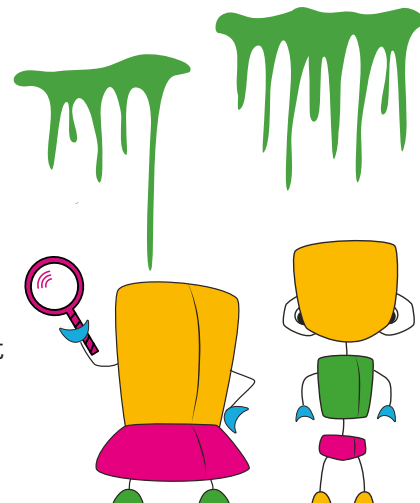
Lägg till i lista (våning + " Slime")

Om (krossade minneskort)

Lägg till i lista (våning + " Krossade minneskort")

Eftersom det här är en lista med text (som också kan kallas *strängar*) så betyder plustecken (+) inte att det ska ske en uträkning, utan strängarna slås samman till en längre sträng. Ett exempel kan vara "20 Smulor".

Just denna dagen hittar Random fyra spår: Smulor på våning 20 och 55, slime på våning 71 och krossade minneskort på våning 18. För varje spår fylls Randoms lista med strängar!



Frågor:

- 1. Vilket spår kommer att ligga först i Ramdoms lista?
- 2. Vilket spår kommer att ligga sist i Randoms lista?
- 3. Kan du hitta på ett nytt spår att lägga till i listan, genom att lägga till ett villkor i kodsnutten?

Vad har detta med datorer och kod att göra?

En *lista* är en av de datastrukturer (vi berättade om datastrukturer i Utmaning 2) som används mest vid programmering. De är bra för att hålla reda på flera saker som hänger ihop, till exempel misstänkta för ett specifikt brott, namn på robotdetektiver eller de senaste temperaturmätningarna på olika våningar i Nordopolis.

Som du kanske har sett så finns listor i Scratch, och även i de flesta andra programmeringsspråk. Då kallas det ibland för en *array*.

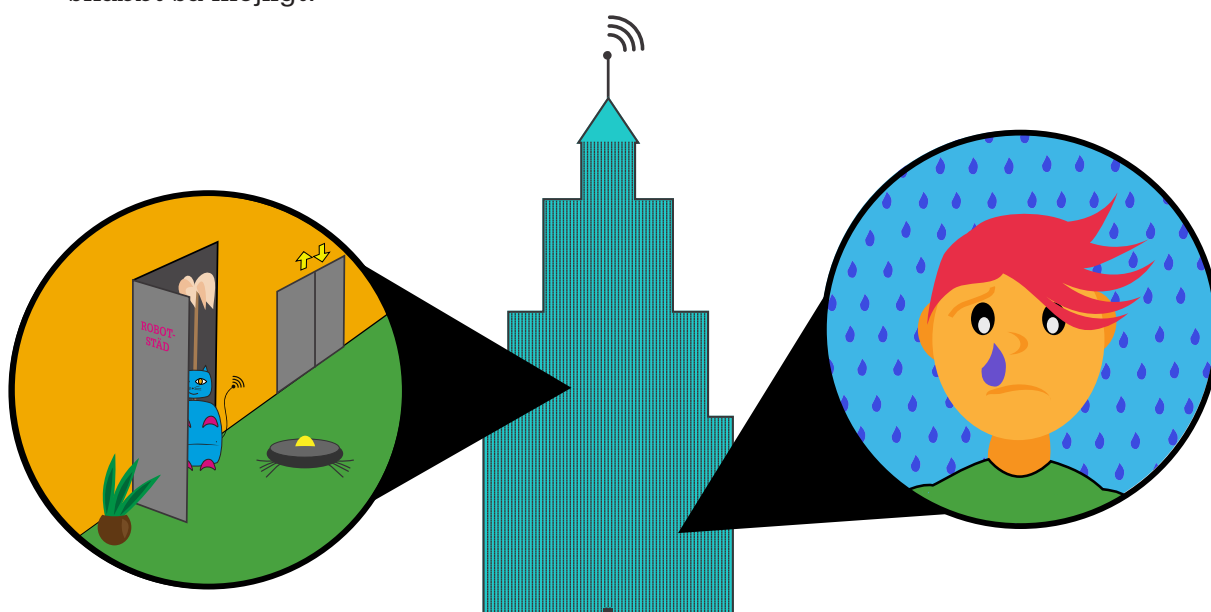


Utmaning 10:

Var är robotkatten?

En robotkatt är försvunnen någonstans i Nordopolis och en pojke är jätteledsen! Roboten Else får i uppdrag att hitta den.

När en robotkatt blir rädd gömmer den sig i en städskrubb och stannar där tills någon aktiverar kattens trygghetsprogram. Det gäller att hitta katten så snabbt så möjligt, om den är rädd i mer än 24 timmar måste minnet återställas. Else behöver därför använda en algoritm för att hitta katten så snabbt så möjligt.



Det finns två städskrubbar på varje våningsplan, båda är nära en hiss. Dessa kallas A och B i koden. Robotarna måste leta i alla städskrubbar tills de har hittat robotkatten.

Katten sågs senast på våning 49, men den kan ha följt med hissarna till andra våningsplan. Att ta hissen mellan två våningar tar oftast 1 minut, men om det är kö och mycket folk kan det ta 10 minuter. Att ta sig mellan städskrubbarna på ett våningsplan tar 7 minuter.

Här ser du två olika algoritmer som Else kan använda för att leta efter katten. Algoritmen avslutas när Else hittar katten. Nu gäller det att välja den algoritm som går snabbast att utföra!



Algoritm 1:

våningAttUndersöka = 48

repetera tills våningAttUndersöka = 1

undersökStädskrubbA

undersökStädskrubbB

våningAttUndersöka = våningAttUndersöka - 1

våningAttUndersöka = 49

repeterat tills våningAttUndersöka = 200

undersökGömställeA

undersökGömställeB

våningAttUndersöka = våningAttUndersöka + 1



Algoritm 2:

våningAttUndersöka = 1

repetera tills våningAttUndersöka = 200

undersökStädskrubbA

våningAttUndersöka = våningAttUndersöka + 1

repetera tills våningAttUndersöka = 1

undersökStädskrubbB

våningAttUndersöka = våningAttUndersöka - 1

Frågor:

- 1. Beskriv med egna ord vad roboten gör i de olika algoritmerna. Beskriv även hur algoritmerna är lika och olika.
- 2. Katten gömmer sig i städskrubb A på våning 40. Hur lång tid tar det för Else att hitta den med algoritm 1: Om hissen alltid tar 1 minut? Om hissen alltid tar 10 minuter? Gör en tabell och skriv in båda dina uträkningar!
- 3. Om Else istället använder algoritm 2, hur lång tid tar det då att hitta katten? Skriv i tabellen och jämför svaren med förra uppgiften. Vilken algoritm var snabbast om hissen tog 1 minut? 10 minuter?
- 4. Fundera ut en egen algoritm för att leta efter katten. Skriv algoritmen med pseudokod. Fyll sedan i tabellen och se hur lång tid det tar för Else att hitta katten med den nya algoritmen!

Vad har detta med datorer och kod att göra?

Du kan ofta skapa flera olika algoritmer för att lösa ett problem. Olika algoritmer har olika fördelar. Ibland vill du använda den som går snabbast, och då kan du undersöka hur lång tid det tar i snittfallet, i bästa fallet och i värsta fallet. I exemplet med kattsökningen beror vad som är bästa och värsta fallet både på var katten gömmer sig och hur lång tid hissarna tar.

</Slut>

Else och Random har nu slutfört alla tre nivåer på detektivskolan. De har klarat alla utmaningar och ska få sina diplom som visar att de tillhör Nordopolis detektivrobotar.

På ceremonin som hålls på våning 200, högst upp i Nordopolis med utsikt över hela tornstaden, står Else och Random på rad tillsammans med tre andra robotar som också klarat utbildningen: While, Repeat och If.

En efter en tilldelas robotarna en medalj och diplom. Där står det - och här kan även du fylla i ditt namn när du har slutfört utmaningarna:

Grattis!

_____ har slutfört Nordopolis
datalogiska utmaningar. Lycka till
med att fortsätta klura och koda!

