

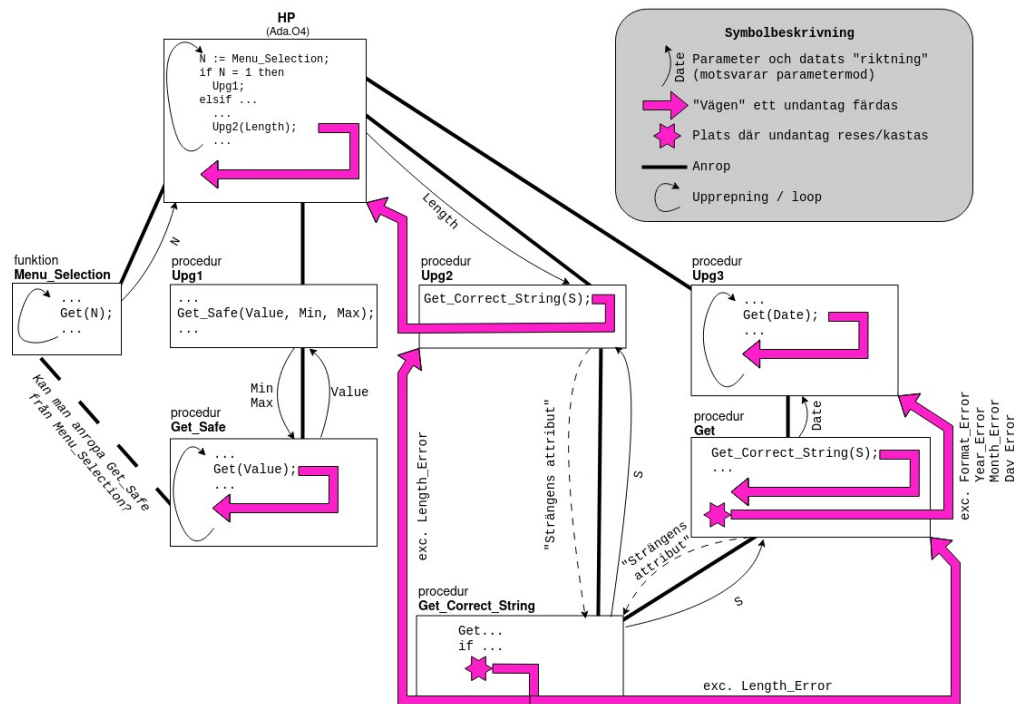
Ada.O4: Undantagshantering

Mål

Du skall efter denna uppgift kunna:

- Fånga undantag på korrekt ställe i koden.
- Resa undantag från korrekt ställe.
- Skapa egna undantag.
- Upptäcka radslut med `End_Of_Line`.

Uppgift



I denna uppgift ska du komplettera samt utöka ett givet program som finns på filen `test_exceptions.adb`. Bilden ovan visar en översiktsbild över programmets olika delar samt hur undantag ska hanteras.

Det givna programmet skriver ut en meny med 4 val. Användaren kan här upprepat utföra valen i menyn genom att mata in heltal:

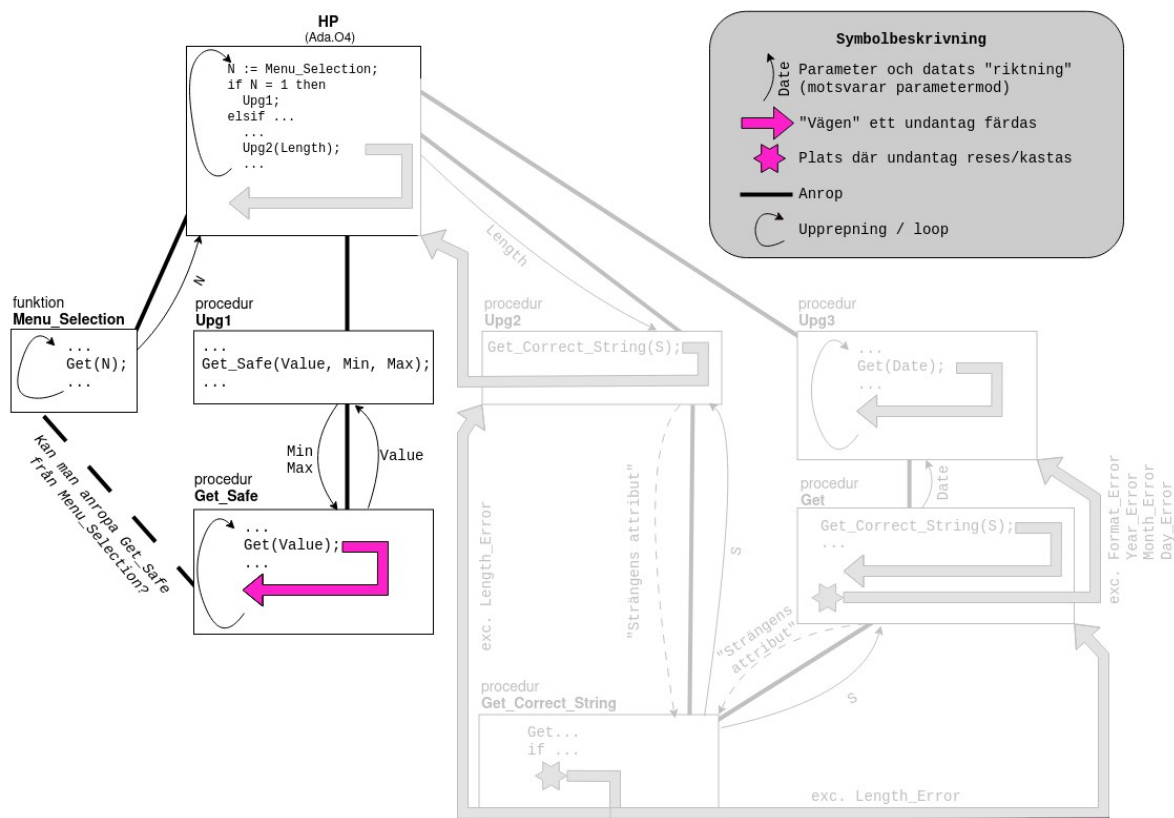
1. Programmet skall felhantera heltalsinmatning där användaren väljer minsta och största värde (**del 1**).
2. Programmet skall felhantera stränginmatning där användaren väljer stränglängden (**del 2**).
3. Programmet ska felhantera datuminmatning (**del 3**).
4. Programmet ska avslutas.

I det givna huvudprogrammet finns redan anropen till de underprogram du skall skapa. Din uppgift är att skapa själva underprogrammen samt lägga till undantagshantering där detta behövs.

KRAV (detta gäller samtliga delar): Alla huvud- och underprogram får ha som mest ca 20 rader kod mellan översta "begin" och nedersta "end". Du kommer alltså antagligen behöva lägga till fler underprogram än de som är specificerade i uppgiften. Tänk över uppdelningen av underprogram, vissa underprogram kanske kan anropas från flera ställen i koden.

Del 1:

Get för heltal (i Ada.Integer_Text_IO) kan kasta/resa undantag så som till exempel Data_Error, Constraint_Error, etc. I denna deluppgift är det tanken att du ska fånga ett sådant för att kunna göra bättre felhantering.



Skriv ett underprogram Get_Safe som har tre heltalsparametrar Value, Min och Max. Underprogrammet skall låta användaren mata in Value via tangentbordet med felhantering enligt nedan. I bilden ovan är relevanta underprogram och undantag markerade i färg.

Value ska få matas in tills denna är som minst Min och som mest Max. Programmet skall även felhantera Value så att användaren kan mata in felaktiga datatyper utan att varken hela programmet krashar eller underprogrammet avslutas.

Du kan anta att användaren alltid matar in korrekta värden på Min och Max, ingen felhantering av dessa krävs. Användaren matar alltså endast in heltal och Min är garanterat mindre än eller lika stort som Max.

Körexempel 1:

1. Felkontrollerad heltalsinläsning
 2. Längdkontrollerad stränginläsning
 3. Felkontrollerad datuminläsning
 4. Avsluta programmet
- Mata in N: **4**
- Programmet avslutas.

Körexempel 2:

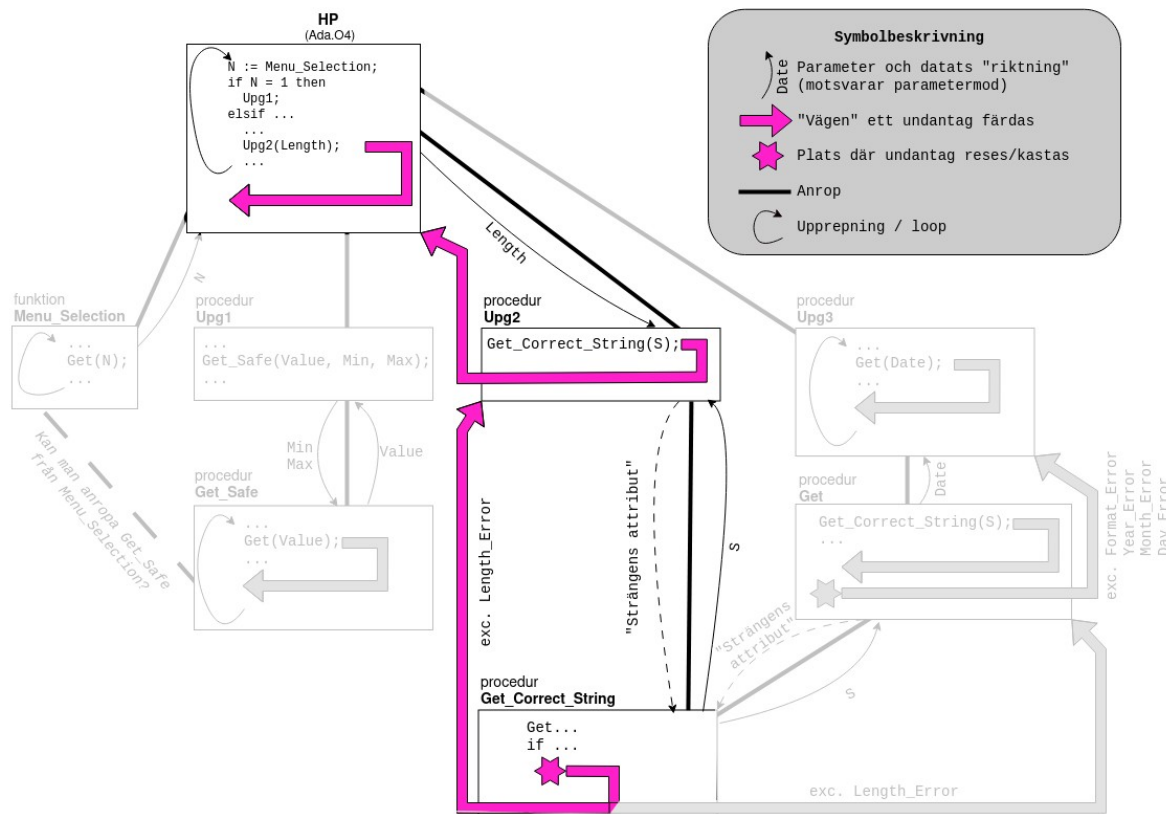
```
1. Felkontrollerad heltalsinläsning
2. Längdkontrollerad stränginläsning
3. Felkontrollerad datuminläsning
4. Avsluta programmet
Mata in N: 1
Mata in Min och Max: -405 2
Mata in värde (-405 - 2): -406
För litet värde. Mata in värde (-405 - 2): 3
För stort värde. Mata in värde (-405 - 2): -70
Du matade in heltalet -70.
1. Felkontrollerad heltalsinläsning
2. Längdkontrollerad stränginläsning
3. Felkontrollerad datuminläsning
4. Avsluta programmet
Mata in N: 4
Programmet avslutas.
```

Körexempel 3:

```
1. Felkontrollerad heltalsinläsning
2. Längdkontrollerad stränginläsning
3. Felkontrollerad datuminläsning
4. Avsluta programmet
Mata in N: 1
Mata in Min och Max: 1 102
Mata in värde (1 - 102): sju
Fel datatyp. Mata in värde (1 - 102): !
Fel datatyp. Mata in värde (1 - 102): MassorMedText
Fel datatyp. Mata in värde (1 - 102): hejsan3
Fel datatyp. Mata in värde (1 - 102): 103
För stort värde. Mata in värde (1 - 102): 0
För litet värde. Mata in värde (1 - 102): 102
Du matade in heltalet 102.
1. Felkontrollerad heltalsinläsning
2. Längdkontrollerad stränginläsning
3. Felkontrollerad datuminläsning
4. Avsluta programmet
Mata in N: 1
Mata in Min och Max: 1000 2004
Mata in värde (1000 - 2004): text som är lång
Fel datatyp. Mata in värde (1000 - 2004): 3
För litet värde. Mata in värde (1000 - 2004): 3
För litet värde. Mata in värde (1000 - 2004): 1000
Du matade in heltalet 1000.
1. Felkontrollerad heltalsinläsning
2. Längdkontrollerad stränginläsning
3. Felkontrollerad datuminläsning
4. Avsluta programmet
Mata in N: 4
Programmet avslutas.
```

Del 2:

Ett undantag som inte fångas kommer passera igenom alla "tidigare" underprogram mellan platsen där undantaget kastats och huvudprogrammet. I denna uppgift är tanken att du ska kasta ett eget undantag och låta detta passera igenom underprogram för att slutligen fångas i huvudprogrammet.



Skriv ett underprogram `Get_Correct_String` som har en strängparameter `S`. Underprogrammet skall låta användaren mata in `S` via tangentbordet med felhantering enligt nedan. I bilden ovan är relevanta underprogram och undantag markerade i färg.

Underprogrammet skall låta användaren mata in en text som kan inledas med godtyckligt många blanktecken och radbrytningar. Underprogrammet skall, likt `Get` för heltal och flyttal, ignorera alla inledande blanktecken och radbrytningar för att sedan läsa in själva texten till `S`.

OBS! Om den inmatade texten är kortare än vad `S` är lång (vi kommer till radslut innan vi fyllt strängen) ska undantaget `Length_Error` kastas/resas av underprogrammet. Underprogrammet ska inte läsa in eller rensa bort det som ligger "efter" den inmatade texten i tangentbordsbufferten (texten kan innehålla blanktecken).

Om `Length_Error` kastas/reses ska huvudprogrammet fånga detta undantag, skriva ut ett felmeddelande och sedan direkt avsluta programmet.

Notera! `Get_Correct_String` som beskrivs ovan har alltså vissa likheter med den `Get_Line` som du redan använt i kursen, men med skillnaden att denna ignorerar inledande blanktecken och radbrytningar samt att den **inte** skall ta bort radslutstecknet efter själva texten. Du kan alltså inte använda `Get_Line` i din lösning.

Körexempel 4:

```
1. Felkontrollerad heltalsinläsning
2. Längdkontrollerad stränginläsning
3. Felkontrollerad datuminläsning
4. Avsluta programmet
Mata in N: 2
Mata in en stränglängd: 5
Mata in en sträng med exakt 5 tecken: hello
Du matade in strängen hello.
1. Felkontrollerad heltalsinläsning
2. Längdkontrollerad stränginläsning
3. Felkontrollerad datuminläsning
4. Avsluta programmet
Mata in N: 4
Programmet avslutas.
```

Körexempel 5:

```
1. Felkontrollerad heltalsinläsning
2. Längdkontrollerad stränginläsning
3. Felkontrollerad datuminläsning
4. Avsluta programmet
Mata in N: 2
Mata in en stränglängd: 10
Mata in en sträng med exakt 10 tecken: Pasta food
Du matade in strängen Pasta food.
1. Felkontrollerad heltalsinläsning
2. Längdkontrollerad stränginläsning
3. Felkontrollerad datuminläsning
4. Avsluta programmet
Mata in N: 2
Mata in en stränglängd: 7
Mata in en sträng med exakt 7 tecken:
```

Hejsan!

```
Du matade in strängen Hejsan!.
1. Felkontrollerad heltalsinläsning
2. Längdkontrollerad stränginläsning
3. Felkontrollerad datuminläsning
4. Avsluta programmet
Mata in N: 2
Mata in en stränglängd: 3
Mata in en sträng med exakt 3 tecken: hoppla hejsan
Du matade in strängen hop.
1. Felkontrollerad heltalsinläsning
2. Längdkontrollerad stränginläsning
3. Felkontrollerad datuminläsning
4. Avsluta programmet
Mata in N: 2
Mata in en stränglängd: 5
Mata in en sträng med exakt 5 tecken: hej
För få inmatade tecken!
```

Körexempel 6 (användarens inmatning understruken för att visa blanktecken):

1. Felkontrollerad heltalsinläsning
2. Längdkontrollerad stränginläsning
3. Felkontrollerad datuminläsning
4. Avsluta programmet

Mata in N: 2

Mata in en stränglängd: 3

Mata in en sträng med exakt 3 tecken: ____

tja

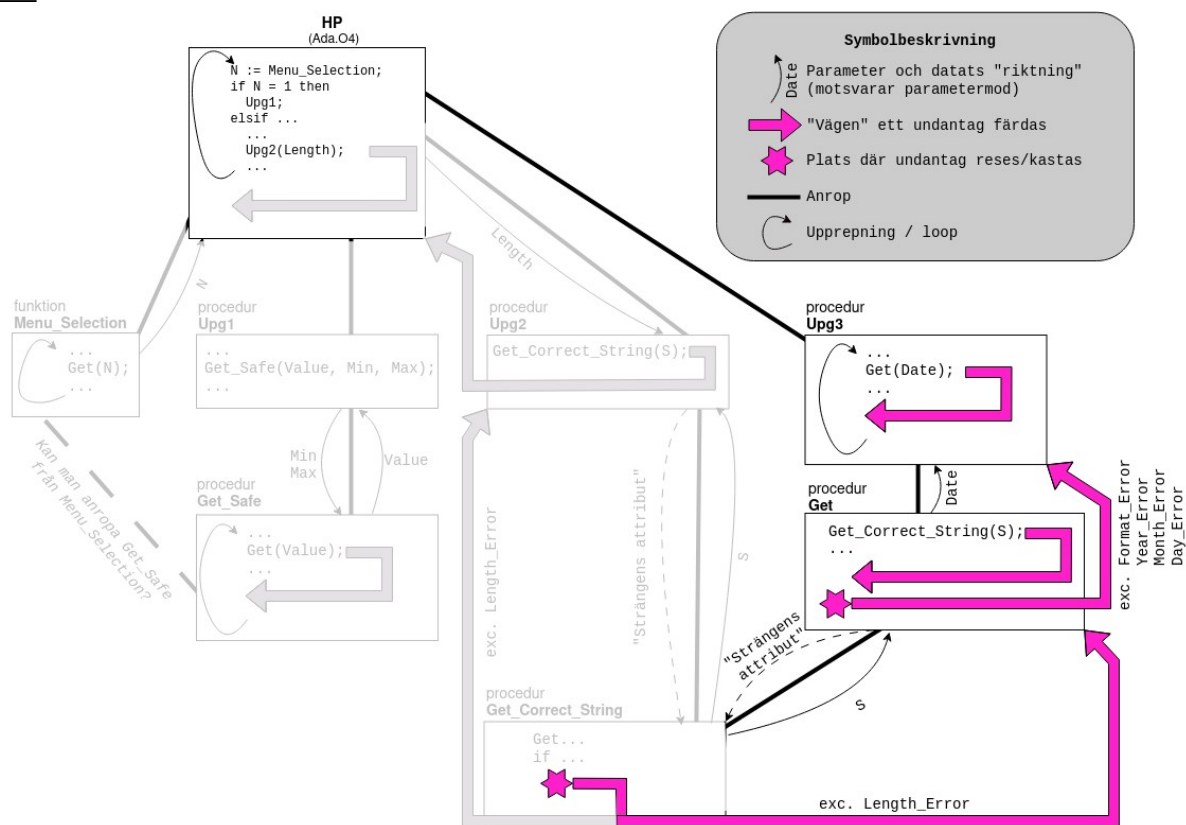
Du matade in strängen tja.

1. Felkontrollerad heltalsinläsning
2. Längdkontrollerad stränginläsning
3. Felkontrollerad datuminläsning
4. Avsluta programmet

Mata in N: 4

Programmet avslutas.

Del 3:



I bilden ovan är relevanta underprogram och undantag markerade i färg. Du skall skapa:

1. en postdatatyp `Date_Type` som består av tre heltal. Denna datatyp representerar ett datum angivet med år, månad och dag.
2. ett underprogram `Put` som har en `Date_Type`-parameter `Item`. Underprogrammet skall skriva ut `Item` på formatet YYYY-MM-DD (mer information nedan).
3. ett underprogram `Get` som har en `Date_Type`-parameter `Item`. Underprogrammet skall låta användaren mata in ett datum enligt formatet YYYY-MM-DD (mer information nedan). `Get` skall låta användaren mata in **ett** datum och felkontrollera detta enligt:
 - Om datumformatet inte följs ska undantaget `Format_Error` resas/kastas. Detta gäller även om användaren matar in ett för kort datum (ex. 2001-2-03).
 - Om året är mindre än 1532 eller större än 9000 skall undantaget `Year_Error` kastas/resas.
 - Om månaden är felaktig skall undantaget `Month_Error` kastas/resas.
 - Om dagen är felaktig (utanför månadens antal dagar med hänsyn till skottår) skall undantaget `Day_Error` kastas/resas. Beskrivning av skottår finns nedan.

Ett korrekt datum ska "skickas tillbaka" via `Item`. Vid inkorrekt datum ska underprogrammet `Upg3` hantera de undantag som reses/kastas, skriva ut felmeddelande och sedan anropa `Get` igen tills att användaren matar in ett korrekt datum.

KRAV: `Get` ska anropa underprogrammet ifrån **del 2** för att läsa in datumet som en 10 tecken lång sträng. Du får **inte** modifiera underprogrammet från **del 2** för att lösa detta problem.

Datumformatet YYYY-MM-DD är exakt 10 tecken långt där YYYY är exakt fyra siffror som motsvarar år, MM är exakt två siffror som motsvarar månad och DD är exakt två siffror som motsvarar dag. YYYY och MM respektive MM och DD skall separeras med exakt ett bindestreck.

Info om månader och skottår:

Månad: Jan Feb Mar Apr Maj Jun Jul Aug Sep Okt Nov Dec
Dagar: 31 28/29 31 30 31 30 31 31 30 31 30 31

Ett år är ett skottår om året är jämnt delbart på 4 men inte jämnt delbart med 100. Dock är år som är jämnt delbara med 400 alltid skottår. Om det är skottår så har februari 29 dagar istället för 28.

Exempel:

År 3: Inte skottår (ej jämnt delbart med 4)
År 4: Skottår (jämnt delbart med 4)
År 200: Inte skottår (jämnt delbart med 4 och 100)
År 400: Skottår (jämnt delbart med 400)
År 2000: Skottår (jämnt delbart med 400)
År 2100: Inte skottår (jämnt delbart med 4 och 100)

Körexempel 7:

1. Felkontrollerad heltalsinläsning
2. Längdkontrollerad stränginläsning
3. Felkontrollerad datuminläsning
4. Avsluta programmet

Mata in N: **3**

Mata in ett datum: **AAAA-02-03**

Felaktigt format! Mata in ett datum: **1997-BB-10**

Felaktigt format! Mata in ett datum: **2000-05-CC**

Felaktigt format! Mata in ett datum: **2000.05-10**

Felaktigt format! Mata in ett datum: **2010-01!07**

Felaktigt format! Mata in ett datum: **1234567890**

Felaktigt format! Mata in ett datum: **2000-01-4**

Felaktigt format! Mata in ett datum:

2000-1-04

Felaktigt format! Mata in ett datum: **199-01-05**

Felaktigt format! Mata in ett datum: **AABBCC**

Felaktigt format! Mata in ett datum: **AAAA-BB-CC2000-04-07**

Felaktigt format! Mata in ett datum: Du matade in 2000-04-07

1. Felkontrollerad heltalsinläsning
2. Längdkontrollerad stränginläsning
3. Felkontrollerad datuminläsning
4. Avsluta programmet

Mata in N: **4**

Programmet avslutas.

Körexempel 8:

```
1. Felkontrollerad heltalsinläsning
2. Längdkontrollerad stränginläsning
3. Felkontrollerad datuminläsning
4. Avsluta programmet
Mata in N: 3
Mata in ett datum: 1531-05-04
Felaktigt år! Mata in ett datum: 9001-05-04
Felaktigt år! Mata in ett datum: 1532-00-04
Felaktig månad! Mata in ett datum: 9000-13-04
Felaktig månad! Mata in ett datum: 1990-01-00
Felaktig dag! Mata in ett datum: 1990-12-32
Felaktig dag! Mata in ett datum: 1990-01-32
Felaktig dag! Mata in ett datum: 1990-02-29
Felaktig dag! Mata in ett datum: 1990-03-32
Felaktig dag! Mata in ett datum: 1990-04-31
Felaktig dag! Mata in ett datum: 1990-05-32
Felaktig dag! Mata in ett datum: 1990-06-31
Felaktig dag! Mata in ett datum: 1990-07-32
Felaktig dag! Mata in ett datum: 1990-08-32
Felaktig dag! Mata in ett datum: 1990-09-31
Felaktig dag! Mata in ett datum: 1990-10-32
Felaktig dag! Mata in ett datum: 1990-11-31
Felaktig dag! Mata in ett datum: 1990-01-31
Du matade in 1990-01-31
1. Felkontrollerad heltalsinläsning
2. Längdkontrollerad stränginläsning
3. Felkontrollerad datuminläsning
4. Avsluta programmet
Mata in N: 3
Mata in ett datum: 1990-02-28
Du matade in 1990-02-28
1. Felkontrollerad heltalsinläsning
2. Längdkontrollerad stränginläsning
3. Felkontrollerad datuminläsning
4. Avsluta programmet
Mata in N: 3
Mata in ett datum: 1990-03-31
Du matade in 1990-03-31
1. Felkontrollerad heltalsinläsning
2. Längdkontrollerad stränginläsning
3. Felkontrollerad datuminläsning
4. Avsluta programmet
Mata in N: 3
Mata in ett datum: 1990-04-30
Du matade in 1990-04-30
1. Felkontrollerad heltalsinläsning
2. Längdkontrollerad stränginläsning
3. Felkontrollerad datuminläsning
4. Avsluta programmet
Mata in N: 3
Mata in ett datum: 1990-05-31
Du matade in 1990-05-31
1. Felkontrollerad heltalsinläsning
2. Längdkontrollerad stränginläsning
3. Felkontrollerad datuminläsning
4. Avsluta programmet
Mata in N: 4
Programmet avslutas.
```

Körexempel 9:

1. Felkontrollerad heltalsinläsning
2. Längdkontrollerad stränginläsning
3. Felkontrollerad datuminläsning
4. Avsluta programmet

Mata in N: **3**

Mata in ett datum: **1990-06-30**

Du matade in 1990-06-30

1. Felkontrollerad heltalsinläsning
2. Längdkontrollerad stränginläsning
3. Felkontrollerad datuminläsning
4. Avsluta programmet

Mata in N: **3**

Mata in ett datum: **1990-07-31**

Du matade in 1990-07-31

1. Felkontrollerad heltalsinläsning
2. Längdkontrollerad stränginläsning
3. Felkontrollerad datuminläsning
4. Avsluta programmet

Mata in N: **3**

Mata in ett datum: **1990-08-31**

Du matade in 1990-08-31

1. Felkontrollerad heltalsinläsning
2. Längdkontrollerad stränginläsning
3. Felkontrollerad datuminläsning
4. Avsluta programmet

Mata in N: **3**

Mata in ett datum: **1990-09-30**

Du matade in 1990-09-30

1. Felkontrollerad heltalsinläsning
2. Längdkontrollerad stränginläsning
3. Felkontrollerad datuminläsning
4. Avsluta programmet

Mata in N: **3**

Mata in ett datum: **1990-10-31**

Du matade in 1990-10-31

1. Felkontrollerad heltalsinläsning
2. Längdkontrollerad stränginläsning
3. Felkontrollerad datuminläsning
4. Avsluta programmet

Mata in N: **3**

Mata in ett datum: **1990-11-30**

Du matade in 1990-11-30

1. Felkontrollerad heltalsinläsning
2. Längdkontrollerad stränginläsning
3. Felkontrollerad datuminläsning
4. Avsluta programmet

Mata in N: **3**

Mata in ett datum: **1990-12-31**

Du matade in 1990-12-31

1. Felkontrollerad heltalsinläsning
2. Längdkontrollerad stränginläsning
3. Felkontrollerad datuminläsning
4. Avsluta programmet

Mata in N: **4**

Programmet avslutas.

Körexempel 10:

```
1. Felkontrollerad heltalsinläsning
2. Längdkontrollerad stränginläsning
3. Felkontrollerad datuminläsning
4. Avsluta programmet
Mata in N: 3
Mata in ett datum: 2000-02-28
Du matade in 2000-02-28
1. Felkontrollerad heltalsinläsning
2. Längdkontrollerad stränginläsning
3. Felkontrollerad datuminläsning
4. Avsluta programmet
Mata in N: 3
Mata in ett datum: 2000-02-29
Du matade in 2000-02-29
1. Felkontrollerad heltalsinläsning
2. Längdkontrollerad stränginläsning
3. Felkontrollerad datuminläsning
4. Avsluta programmet
Mata in N: 3
Mata in ett datum: 2000-02-30
Felaktig dag! Mata in ett datum: 2000-02-31
Felaktig dag! Mata in ett datum: 2100-02-29
Felaktig dag! Mata in ett datum: 2004-02-29
Du matade in 2004-02-29
1. Felkontrollerad heltalsinläsning
2. Längdkontrollerad stränginläsning
3. Felkontrollerad datuminläsning
4. Avsluta programmet
Mata in N: 3
Mata in ett datum: 2100-02-30
Felaktig dag! Mata in ett datum: 2100-02-31
Felaktig dag! Mata in ett datum: 2004-02-32
Felaktig dag! Mata in ett datum: 1875-05-14
Du matade in 1875-05-14
1. Felkontrollerad heltalsinläsning
2. Längdkontrollerad stränginläsning
3. Felkontrollerad datuminläsning
4. Avsluta programmet
Mata in N: 4
Programmet avslutas.
```