**Uppgift - Under arbete**

Fyll i formuläret, välj sedan passande knapp längst ned.

| | |
|---|---|
| **Titel** | OU5 - Automat |
| **Sista inlämning** | 2016-03-22 17:00 |
| **Antal tillåtna återinlämningar** | Obegränsat |
| **Tillåt återlämningar till** | 2016-06-10 17:00 |
| **Status** | Under arbete |
| **Betygsskala** | Poäng (max 10,0) |
| **Ändrad av lärare** | 2016-02-25 09:00 |

**Instruktioner**

## Syfte

Syftet med denna obligatoriska uppgift är att du ska:

- öva på att använda automater för att lösa problem
- sätta samman flera olika datatyper och välja deras representation för att lösa ett problem

De FSR (lärmål) som examineras (helt eller delvis) med denna uppgift är:

- beskriva och använda sig av abstrakta datatyper och algoritmer (FSR 1),
- implementera lösningen på ett problem i form av ett program i programspråket C inklusive att välja en lämplig representation för de ingående abstrakta datatyperna (FSR 7),
- använda en algoritm för att avgöra medlemskap i ett språk definierat av en kontextfri grammatik (FSR 9),
- utifrån en frågeställning göra relevanta designbeslut under arbetet med att lösa ett problem utifrån sin kunskap om hur struktur-, tids- och rumsaspekter påverkar kvalitet hos program (FSR 10)

## Uppgift

I denna uppgift ska du implementera en tillståndsmaskin/automat. Automaten du implementerar kommer vara rätt lik en Push-Down Automat (PDA) men ha annan uttryckskraft. Din automat ska dels kontrollera om ett givet uttryck är skrivet i korrekt omvänd polsk notation och om så är fallet beräkna värdet av uttrycket.

## Krav

- Du ska genomföra uppgiften **enskilt** och den ska vara inlämnad via Cambro senast **tisdag 22/3 kl. 17.00.**
- Din lösning ska fungera enligt anvisningarna under rubriken Genomförande och de krav som ställs på datatyperna och programmets beteende ska vara uppfyllda.
- Ditt arbete ska redovisas via en skriftlig rapport **i pdf-format** som ska innehålla följande:
  - En framsida enligt denna mall (word- eller pdf-format).
  - En bra beskrivning av hur alla intressanta delar av din lösning fungerar. Vilka datatyper har du skapat, hur har du valt att representera dem och hur ser deras gränsyta ut? Hur samarbetar de?
    - Använd kursboken av Janlert/Wiberg som inspiration när du beskriver dina datatyper.
  - Testköningar som visar att din kod fungerar. Dokumentera testkörningarna noggrant på detta sätt:
    - Beskriv bakgrunden, vilken automat testkör du (visa med en bild på automaten eller skriv ut tabellen med transitioner).
    - Beskriv testfallet, vad ska du testa och vilket svar förväntar du dig?
    - Resultatet, dvs själva testet, vad skrevs ut av programmet?
    - Slutsats, fungerade ditt program som tänkt?
  - Berätta om ditt arbete med laborationen. Vad tycker du var lätt, vad var svårt, hur löste du de problem som dök upp? Har du synpunkter på själva uppgiften?
- Du ska också lämna in välstrukturerad och kommenterad kod.
  - Ta inte med källkodsfiler som inte ingår i lösningen.
  - Man ska kunna kompilera din kod med kommandot `gcc -std=c99 -Wall *.c` utan att få varningar.
  - Koden ska inte innehålla minnesläckor.
- Uppgiften ska laddas upp som en zip-fil som innehåller alla filer inklusive rapporten.

## Genomförande

Din automat ska dels kontrollera om ett givet uttryck är skrivet i korrekt omvänd polsk notation och om så är fallet beräkna värdet av uttrycket. Bara positiva heltal måste stödjas (int) och de binära operationerna: + - * och /. Om uttrycket ej är korrekt ska texten "Invalid expression" skrivas ut annars skrivs värdet av uttrycket ut. Notera att du måste skriva uttrycket inom " och ", annars tolkar systemet din sekvens av tecken som en sekvens av parametrar till programmet istället för en enda parameter.

**Exempel på hur körningar av programmet ska se ut:**

```
>automat "1"
1
>automat "1 2 3+×0 5+÷"
1
>automat "fjdslajf341.4+"
Invalid expression
>automat "3+4*4"
Invalid expression
>automat "3 4+4*"
28
```

Din datatyp för automaten **ska** vara utformad så att godtyckliga tillstånd och transitioner kan läggas till, man ska alltså i princip kunna använda den för att lösa andra problem också. Ett **krav** på automaten är att den maximalt får läsa ett tecken i taget från inputsträngen.

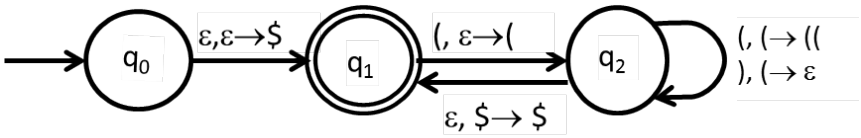
För transitionerna **ska** en funktion tillhandahållas som avgör om villkoret för transitionen är uppfylld (utifrån den aktuella bokstaven i inputsträngen, vad som finns på stacken och om stacken är tom) samt en funktion som utförs då transitionen utnyttjas av automaten (som tex kan förändra stacken utifrån värdet på stacken och den aktuella bokstaven från inputsträngen).

### Datatyper

De datatyper som du skapar ska läggas i sina egna c- och h-filer och vara kommenterade på samma sätt de dataypsfiler du sett hittills på kursen. Det ska vara lätt för en annan person att ta dina filer och skriva ett eget program som använder dem. Tänk på hur du skulle vilja att rapport och kommentarer var skrivna om du skulle ta över en kompis kod!

## Frivillig utökning

För att visa att din automatdatatyp är generell så ska du skapa en automat för parentesmatchning. Du ska alltså implementera denna automat och visa med tester att din implementation fungerar.




**Exempel på hur körningar av programmet ska se ut:**

```
>automat ""
Accepted
>automat "(())"
Accepted
>automat "(()("
Invalid expression
```

## Tips

En del av uppgiften handlar om att du ska välja en så bra datareperesentation som möjligt och det finns flera olika sätt att skapa datastrukturer för en automat. Två huvudalternativ är om man ser automaten som en graf eller om man arbetar med en tabellmodell. Nedan ger vi korta beskrivningar av de datatyper (förutom en stack som behövs i båda alternativen) som vi tror att du kommer behöva fundera kring i denna uppgift:

- **Automat som graf**. En automat består av ett antal tillstånd, en stack och något som håller reda på aktuellt

tillstånd och aktuell position i input. Här ska man kunna lägga till tillstånd och också starta en körning av automaten på en given sträng. Körningen startar i starttillståndet som håller reda på sina övergångar och vilka tillstånd man kan nå från det. Man vandrar sedan fram i grafen tills dess man läst all input och befinner man sig då i ett accepterande tillstånd så kan man godkänna strängen. Till hjälp behöver man (minst) dessa datatyper):

- **Tillstånd**, ett tillstånd måste veta om det är ett accepterande tillstånd eller ej och det måste också veta vilka möjliga transitioner som finns kopplat till det. Man bör också kunna "be" ett tillstånd om en giltig transtition utifrån aktuell input och aktuellt tecken på stacken.
- **Transition/övergång**, en transition måste veta det aktuella tillståndet, vilket det nästa tillståndet är och vilka funktioner som ska utföras på input (läsa/inte läsa) och stacken (poppa/pusha/inget). Dessa funktioner skickas lämpligen som funktionspekare. Slutligen måste den kunna genomföra övergången och arbeta med input och stack.

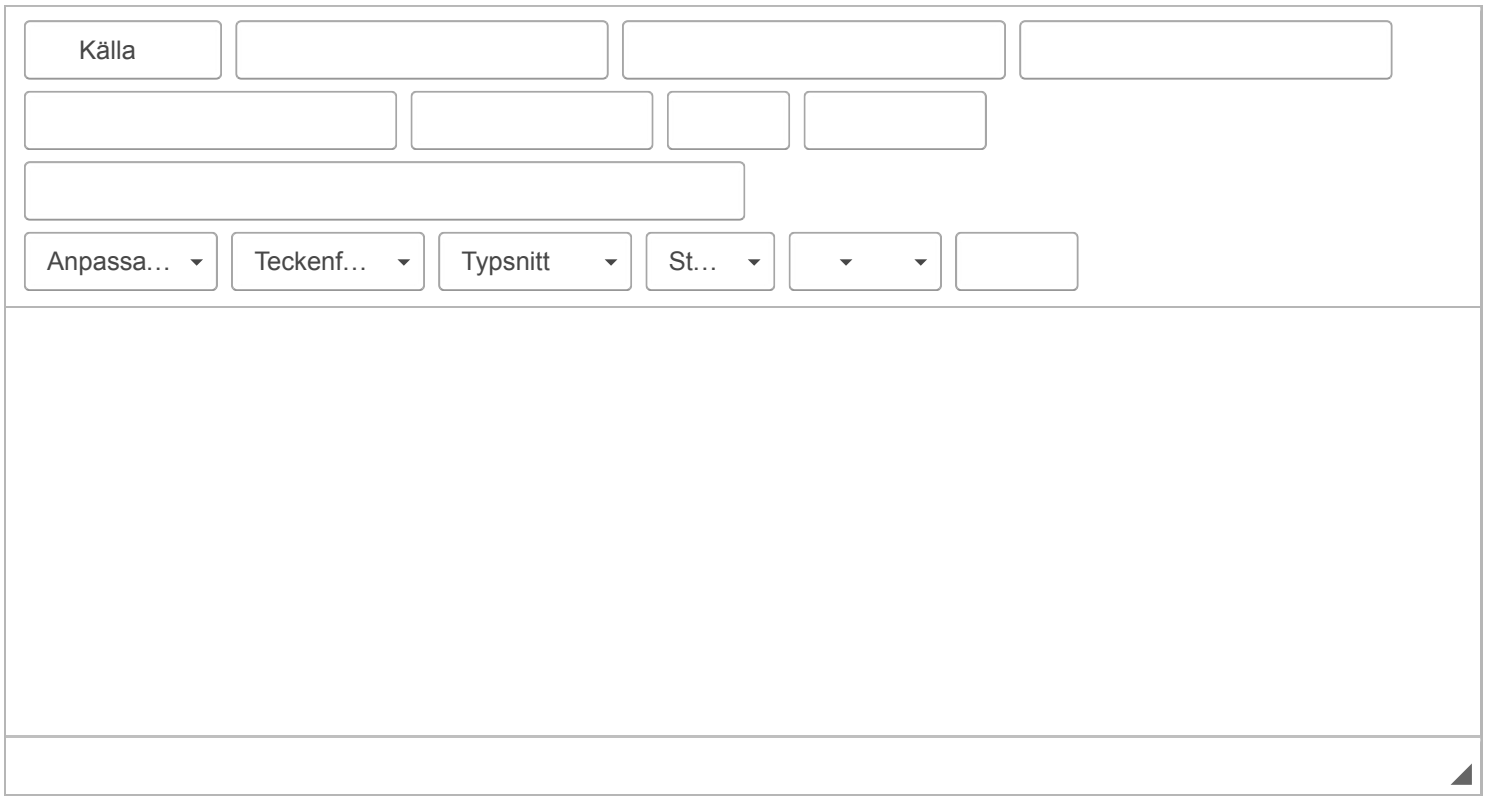
- **Automat med en tabell.** En automat består av ett antal tillstånd, en stack och en tabell där man utifrån ett aktuellt tillstånd, ett inputvärde och ett stackvärde kan slå upp nästa tillstånd och vilka operationer som ska göras på input och stacken. Man gör sedan upprepade slagningar i tabellen tills dess man läst all input och befinner man sig då i ett accepterande tillstånd så kan man godkänna strängen. På något sätt måste information om starttillståndet också lagras. Till hjälp behöver man (minst) dessa datatyper):
  - **Tabell,** Denna tabell har tre nycklar som alla måste stämma (tillstånd, stacksymbol och inputsymbol) och ska som resultat kunna ge nytt tillstånd samt vilka funktioner som ska appliceras på stack och input. Här kanske det behövs speciella datatyper för nyckel och värde?
  - **Tillstånd**, ett tillstånd måste veta om det är ett accepterande tillstånd eller ej.

---

## Inlämning

### Uppgiftens text

Denna uppgift kan lämnas in både genom att använda textrutan nedan och bifoga filer. Skriv i rutan nedan och använda 'Lägg till filer' för att inkludera andra dokument. Spara ofta medan du arbetar.




### Bifogade filer

Inga bifogade filer ännu

Välj en fil från datorn Choose File no file selected