

Ada.O1.1: Mycket grundläggande in- och utmatning

I denna inledande uppgift kommer du att öva på de mest grundläggande operationerna för in- och utmatning.

Mål

Du ska efter denna uppgift ha god insikt i:

- Skillnaden mellan *Get* och *Put* samt *New_Line* och *Skip_Line*.
- Skillnader och likheter mellan de olika varianterna av *Get* och *Put*.
- Skillnaderna i beteende vid olika inmatningar till *Get_Line* och kunna hantera dessa. Notera att *Get_Line* inte helt kan lösa problemet i denna uppgift på egen hand. Här kommer en *if*-sats behövas.

Uppgift

Skriv ett program som hämtar data från tangentbordet enligt exemplen som följer. Det som syns på skärmen när man kör programmet skall stämma till fullo överens med det som står i exemplen. Detta gäller även antal blanktecken mellan ord och tal.

Observera att denna uppgift till stor del är till för att introducera er till hur man hanterar vår datormiljö vad det gäller skrivande, kompilering och körande av program. Om ni inte hinner med allt på laborationstiden är det ingen katastrof. Fortsätt på egen tid med resterande delar så att målen med uppgiften är uppfyllda.

Ett tips kan vara att ni gör en liten del i taget och ser till att den fungerar för de givna körexemplen (se följande sidor). På detta sätt undviker ni att få alla ”fel” på en gång.

KRAV: Observera att du i ditt program **endast får ha en variabel av respektive datatyp**. Det blir alltså bara fyra variabler i programmet. Inga loopar används i detta program.

OBS! Ditt program skall klara av körexempel 1, 2 och 3 (samma program). Dock behöver det inte klara av alla möjliga varianter av indata vilket i princip är omöjligt i detta läge.

TIPS! Skriv först klart programmet så att det klarar av körexempel 1. Modifiera sedan den koden så att den hanterar både körexempel 1 och 2. När koden fungerar för de första två körexemplen bör körexempel 3 fungera utan att du ska behöva modifiera koden.

Programkörningar

Programmet ska ge resultat enligt följande exempel vid olika körningar (användarens indata markeras med *kursiverad* stil). Observera att eventuella inmatningar som är extra skall plockas bort innan nästa inmatning görs.

OBS! Kopiera INTE texten ifrån denna PDF till ditt program då tecknen kan bli felaktiga! OBS!

Körexempel 1:

```
Skriv in ett heltal: 10
Du skrev in talet: 10

Skriv in fem heltal: 12 30 27 13 11
Du skrev in talen: 12 30 27 13 11

Skriv in ett flyttal: 15.20
Du skrev in flyttalet: 15.200

Skriv in ett heltal och ett flyttal: 67 3.141592
Du skrev in heltalet:      67
Du skrev in flyttalet:    3.1416

Skriv in ett tecken: T
Du skrev in tecknet: T

Skriv in en sträng med 5 tecken: Kalle
Du skrev in strängen: Kalle

Skriv in en sträng med 3 tecken: Kul
Du skrev in strängen: Kul

Skriv in ett heltal och en sträng med 5 tecken: -5 Nisse
Du skrev in talet |-5| och strängen |Nisse|.

Skriv in en sträng med 3 tecken och ett flyttal: Ria 1.2
Du skrev in " 1.200" och "Ria".

Skriv in en sträng som är maximalt 5 tecken: Bu
Du skrev in strängen: Bu

Skriv in en sträng som är maximalt 5 tecken: Nysta
Du skrev in strängen: Nysta
```

Körexempel 2 på nästa sidan.

Körexempel 2:

Skriv in ett heltal: **14.72**

Du skrev in talet: 14

Skriv in fem heltal: **10000 3860 27 -18 0 0**

Du skrev in talen: 10000 3860 27 -18 0

Skriv in ett flyttal: **15 Nisse**

Du skrev in flyttalet: 15.000

Skriv in ett heltal och ett flyttal: **7 202.718 Dum text**

Du skrev in heltalet: 7

Du skrev in flyttalet: 202.7180

Skriv in ett tecken: **!K**

Du skrev in tecknet: !

Skriv in en sträng med 5 tecken: **#4?".!**

Du skrev in strängen: #4?".

Skriv in en sträng med 3 tecken: **"HA12**

Du skrev in strängen: "HA

Skriv in ett heltal och en sträng med 5 tecken: **1 Ploja!**

Du skrev in talet |1| och strängen |Ploja|.

Skriv in en sträng med 3 tecken och ett flyttal: **Led 012.3456.23**

Du skrev in "12.346" och "Led".

Skriv in en sträng som är maximalt 5 tecken: **Per ar bra**

Du skrev in strängen: Per a

Skriv in en sträng som är maximalt 5 tecken: **KUL**

Du skrev in strängen: KUL

Körexempel 3 på nästa sida är lite speciell men visar på vad som kommer hända om man matar in lite kluriga saker i programmet. Om du fått körexempel 2 att fungera felfritt ska du inte behöva göra några ändringar i koden för att körexempel 3 ska fungera.

Körexempel 3 (Notera att ÅÄÖ betar sig lite annorlunda. Detta beror på att de internt i datorn består av två Character vardera. Om du vill veta mer om detta får du läsa på själv, sök på UTF-8 på nätet):

Skriv in ett heltal: **10.00 hejsan**

Du skrev in talet: 10

Skriv in fem heltal: **1 -2 3 4**

Du skrev in talen: 1 -2 3 4 **5**

5

Skriv in ett flyttal: **1E-2**

Du skrev in flyttalet: 0.010

Skriv in ett heltal och ett flyttal: **3.14**

Du skrev in heltalet: 3

Du skrev in flyttalet: 0.1400

Skriv in ett tecken: **P**

Du skrev in tecknet: P

Skriv in en sträng med 5 tecken: **Åkeri**

Du skrev in strängen: Åker

Skriv in en sträng med 3 tecken: **Åke**

Du skrev in strängen: Åk

Skriv in ett heltal och en sträng med 5 tecken: **1.23 hejsan**

Du skrev in talet |1| och strängen |23 he|.

Skriv in en sträng med 3 tecken och ett flyttal: **1234.56**

Du skrev in " 4.560" och "123".

Skriv in en sträng som är maximalt 5 tecken: **ÅÄoÖ**

Du skrev in strängen: ÅÄo

Skriv in en sträng som är maximalt 5 tecken:

Du skrev in strängen: