

Laboration 3

Gunnar Landström

| HT24 DT155G_DT027G – Grundläggande datavetenskap |

```
10 def main(): 1 usage
11
12     print("Detta program beräknar SGD för 2 heltal med hjälp av Euklides algoritm")
13     tal_1 = int(input( "Skriv in ett heltal: "))
14     tal_2 = int(input( "Skriv in ett heltal till:"))
15
16     # M = Det större talet
17     # N = Det mindre talet
18
19
20     # Om tal 1 är lika stort eller större än tal 2
21     if tal_1 >= tal_2:
22         M = tal_1
23         N = tal_2
24
25     # Om tal 2 är det större talet
26     else:
```

Programmeringskod skriven i Pycharm Community Edition

Utrustning

Program skrivna med IDE Pycharm Community Edition 2024, information tagna ifrån länkad PDF samt övningar på campus med Karl Pettersson

1.2

Det blir fel för att programmet försöker läsa in int-värden som sträng, den försöker även addera string med int och float för att sedan försöka printa ut resultatet.

Felen beror på att python tolkar alla värden inneslutna av (” ”) som en sträng, en bit av text, men att man sedan vill tolka texten som ett nummervärde och utföra addition.

1.3

I mitt program så frågar jag användaren efter priset på någonting utan moms, sen konverterar jag det till ett int-värde. Momsen är satt till 25% av svaret, och momssatsen multipliceras med användarens inmatning och sparas än en gång som ett int-värde.

Programmet skriver sedan ut: Användarens inmatning, användarens inmatning med moms och till sist enbart momsen.

2.1

Användaren får mata in ett heltal. Programmet tar sedan talet genom en modulus av två och testar ifall värdet inte blir 0.

Om resultatet av input modulus 2 inte blir 0 så resulterar det i ett udda tal.

Annars är resultatet ett jämnt tal.

3.1

Efter att programmet nått sitt slut genom att ta sig ur while loopen så har i värdet 10.

Detta på grund av att sista gången programmet printar i när i är 9, så läggs i + 1 till efter den printen innan det än en gång testas i while loopen om i är mindre än 10.

3.2

Programmet annonserar att det beräknar största gemensamma nämnaren mellan två heltal med hjälp av Euklides algoritm. Sen får användaren mata in två heltal, oberoende av vilken ordning det skrivs i så tilldelas alltid variabeln M det större talet och att variabel N tilldelas det mindre talet.

Sen testas talen emot Euklides algoritm för att sedan printa ut största gemensamma nämnaren.

3.3

Programmet annonserar att det kontrollerar om det årtal som anges är ett skottår.

Det testas om inmatningens modulus av 4 = 0, om inte, skriv ut inte skottår.

Sen testas den om inmatningens modulus av 100 = 0, om inte, skriv ut skottår.

Till sist testas den om inmatningens modulus av 400 = 0, isåfall skriv ut skottår, annars skriv ut inte skottår.

Slutsats

Jag har tack vare denna labb blivit betydligt mer observant när det kommer till att tilldela värden till sina variabler, vilken typ de har som standard och vilka värden som fungerar bra med varandra, alternativt hur man kringgår problem genom att tilldela dem nya värden.

Att man i programmet kan tex Det blir fel för att programmet försöker läsa in int-värden som sträng, den försöker även addera string med int och float för att sedan försöka printa ut resultatet.

2.

Felen beror på att python tolkar alla värden inneslutna av (” ”) som en sträng, en bit av text, men att man sedan vill tolka texten som ett nummervärde och utföra addition.

ta en sträng, göra den till en int, gånga den med en float, spara svaret som en int, för att sedan printa ut resultatet som en sträng igen.