

**Exercise 9: 3rd collection of overheads**

Fill out the tables in the 3rd collection of overheads.

**Exercise 10: Extended tax program**

Change the program from exercise 3 such that it reads an income from the keyboard and prints the following taxes on the screen:

- ambi (Danish: arbejdsmarkedsbidrag): 8 % of the income
- special pension payment (Danish: særligt pensionsbidrag): 1 % of the income
- local taxes (Danish: kommune-, amts- og kirkeskat): 32.8 % of the income after ambi, special pension payment (særligt pensionsbidrag) and personal allowance (personfradrag) have been subtracted
- bottom/basic tax (Danish: bundskat): 7 % of the income after ambi, special pension payment (særligt pensionsbidrag) and personal allowance (personfradrag) have been subtracted
- intermediate tax (Danish: mellemskat): 6 % of the income after ambi, special pension payment (særligt pensionsbidrag) and allowance (bundfradrag) have been subtracted
- top tax (Danish: topskat): 15 % of the income after ambi, special pension payment (særligt pensionsbidrag) and allowance (bundfradrag) have been subtracted

The personal allowance (Danish: personfradrag) was 33,400 kr.

The allowance (Danish: bundfradrag) for intermediate tax was 164,300 kr.

The allowance (Danish: bundfradrag) for top tax was 267,600 kr.

**Exercise 11: Dates**

Write a program that reads a date (three integers) in this millenium and prints the date of the next day.

Example run of the program:

```
Enter year (between 2000 and 2999): 2002
Enter month: 9
Enter day: 30
The day after 30/9 2002 is 1/10 2002
```

Hint:

- Introduce a variable *daysinmonth* for storing the number of days in the month that has been read. Use a switch statement to initialize *daysinmonth*.
- The rules for leap years are as follows: If a year is divisible by 4, it is a leap year. But if the year also is divisible by 100, it is not a leap year. However, if the year also is divisible by 400, it is a leap year. I.e. 1996 is a leap year, 2000 is a leap year, and 2100 is not a leap year.

**Exercise 12: Passwords revisited**

Change the program from exercise 8 such that the program keeps prompting for a password until the user has entered a correct password.