

Programmering 1

Laboration 10: Filhantering, enumerationer

Denna laboration är tänkt att både utföras och **redovisas** under de schemalagda timmar som finns anslagna för ändamålet. Ifall du inte har möjlighet att delta i laborationstillfällena, eller inte hinner slutföra laborationen under den schemalagda tiden måste du redovisa dina resultat i en **skriftlig laborationsrapport** som skickas in via Moodle. Ett obligatoriskt delkrav för att kunna bli godkänd i kursen är att **alla laborationer är godkända**.

Obs: Denna gång kommer alla uppgifter att gås igenom gemensamt under laborationstillfället. Laborationen godkänns alltså automatiskt om du deltar.

1. Enumerationer

Definiera en enumeration (**enum**) för månader (**JAN, FEB, MAR, APR. . .**) och döp denna till **month**. **JAN** ska motsvara heltalet 1, **FEB** ska motsvara 2, osv.

Skapa därefter en räkka **monthNames** som innehåller strängar motsvarande månadernas namn ("**January**", "**February**", . . .)

Skapa slutligen en funktion **get_month_name** som tar en **enum month** som parameter och returnerar en pekare till en sträng som innehåller namnet på månaden i fråga.

get_month_name skall alltså använda sig av **monthNames**. Funktionen och enumerationen skall kunna användas enligt nedanstående modell:

```
enum month aMonth = OCT;
printf("%s\n", get_month_name(aMonth)); // => Utskriften blir
                                         //      October
```

2. Enumerationer och typedef

Modifiera föregående uppgift så att du definierar och använder en typdefinition (**typedef**) för att definiera enumerationen, och kallar denna för **month**.

```
month aMonth = OCT;
printf("%s\n", get_month_name(aMonth)); // Utskriften blir
//      October
```

3. *Kontrollera filstorlek: redirection*

Skriv ett program som använder 'redirection' och räknar antalet bytes i en fil. T.ex. för en fil med storleken 303 bytes bör man få följande resultat:

```
./a.out < testfile  
size is 303 bytes
```

Tips: **getchar()** läser i praktiken in en byte i taget, trots att dess returtyp är **int**
Kontrollera att kommandot **ls -l** ger samma resultat som ditt program!

4. *Kolla filstorlek: namngivna filer*

Utgå från uppgift 3 och skriv ett program som räknar antalet bytes i en namngiven fil, som alltså skall öppnas med hjälp av **fopen()** och stängas med **fclose()**.

Läs in filnamnet med hjälp av **fgets()**

```
Give filename: foo.txt  
Size of foo.txt is 13 bytes
```

```
Give filename: nonexistentfile  
File not found!
```

5. *Kolla filstorlek: programparametrar*

Utgå från uppgift 4 och modifiera det så att filnamnet läses in som en parameter då programmet startas. Se också till att programmet ger ett felmeddelande om det anropas med fel antal parametrar.

```
./a.out foo.txt  
Size of foo.txt is 13 bytes
```

```
./a.out nonexistentfile  
File not found!
```

```
./a.out  
Wrong number of parameters!
```

6. *Kopiera filer*

Utgå från uppgift 5 och skriv ett program som kopierar en fil som anges med hjälp av programparametrar.

```
./a.out foo.txt bar.txt
```

File copied!

```
./a.out nonexistentfile bar.txt
```

File not found!

```
./a.out foo.txt
```

Wrong number of parameters!