Laboration C++ 2 [G]

Skickas in som Uppgift/Assignment 1.

Skriv ett underprogram som sköter inmatning av ett heltal, felhanterat så att inget negativt tal accepteras. Underprogrammet skall ha en parameter (heltalet som man läser in till). Underprogrammet skall returnera sant om inläsningen lyckades på första försöket (användaren matade aldrig in negativt) och falskt annars. Anropa ditt underprogram från huvudprogrammet. Huvudprogrammet skall sköta utskriften av "Du matade in ... och ...".

OBS: Tänk noga över huruvida parametrar är in- eller utdata. Parametrar som endast är indata till en funktion skall deklareras med *const*.

Körexempel 1

```
Mata in ett tal: 4
Du matade in 4 och lyckades på första försöket.
```

Körexempel 2

```
Mata in ett tal: -3
Fel! Mata in igen: -8
Fel! Mata in igen: -1000
Fel! Mata in igen: 0
Du matade in 0 och behövde några försök på dig.
```

Laboration C++ 2 [VG]

Skickas in som Uppgift/Assignment 2.

Givet huvudprogrammet nedan, skriv funktionerna *reset(B), set(B, C)* och *toggle(B)* som fungerar på följande vis.

- reset tar en boolean B som parameter och sätter den till false.
- set tar två booleans B och C och sätter B till C.
- toggle tar en boolean B och sätter den till motsatt värde (true blir false och vice versa).

Alla tre funktionerna skall returnera Bs orginalvärde.

```
1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5    bool B{true};
6    cout << "Före - Efter"     << endl
7         << "   " << reset(B)      << "   " << B << endl
8         << "   " << toggle(B)     << "   " << B << endl
9         << "   " << set(B, false) << "   " << B << endl
10        << "   " << set(B, true)  << "   " << B << endl;
11   return 0;
12 }
```

OBS: Tänk noga över huruvida parametrar är in- eller utdata. Parametrar som endast är indata till en funktion skall deklareras med *const*.

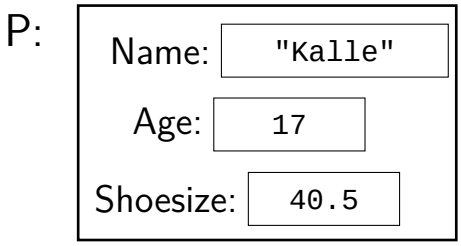
Körexempel 1

```
Före - Efter
   1   0
   0   1
   1   0
   0   1
```

Laboration C++ 3 [G]

Skickas in som Uppgift/Assignment 3.

Definiera en ny typ enligt figuren nedan. Din typ skall vara en *struct* och innehålla de delar som figuren visar. Datat i varje delvariabel visar vilken typ som den delvariabeln skall vara.



Skapa sedan ett huvudprogram som deklararerar en variabel P av din nya typ. Programmet skall sedan göra följande:

1. Låta användaren mata in data som lagras i P.
2. Deklarera en variabel av typen *vector*. Vektorns inre typ skall vara den struct-typ som du definierat.
3. Stoppa in P på första plats i vektorn.
4. Slutligen skall programmet skriva ut datat som ligger lagrat i vektorns första element **(du skall inte använda variabeln P i detta steg)**.

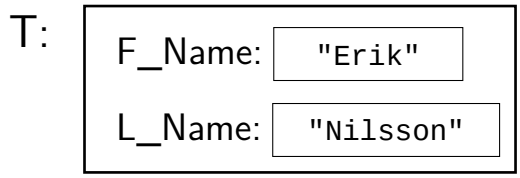
Körexempel 1

```
Mata in namn: Kalle
Mata in ålder: 17
Mata in skostorlek: 40.5

Du matade in Kalle, 17, 40.5
```

Laboration C++ 3 [VG]

Skickas in som Uppgift/Assignment 4.



Skapa ett program där du definierar ett fält (använd typen *vector*) av typen T som visas ovan. T skall vara en *struct*. Låt sedan användaren fylla på med data i denna datastruktur (enligt körexemplet). Inmatningen avslutas med enter och ctrl-d. Slutligen skall programmet gå igenom och räkna hur många personer som hade ett för eller efternamn som började på bokstaven 'E'.

Körexempel 1

```
Mata in personer (avsluta med enter och Ctrl-D):
Erik Nilsson
Nils Eriksson
Evald Erlandsson
Emma Svensson
Viktor Olsson
Enock Jonsson
Jon Enocksson
[ctrl-d]

6 Personer har för- eller efternamn som börjar på 'E'.
```