

# Övning funktioner

## Del 1

1. Skriv en funktion som tar emot ett heltal och skriver ut det. Anropa funktionen från main.
2. Skriv en funktion som tar emot två heltal och skriver ut det största. Anropa funktionen från main.
3. Skriv en funktion som tar emot ett heltal och ett tecken och skriver ut bägge. Anropa funktionen från main.
4. Skriv en funktion som frågar användaren efter ett heltal och returnera talet. Anropa funktionen två gånger från main för att få två tal. Skicka sedan talen till funktionen som skriver ut det största.

## Del 2

Du ska skriva ett program som slumpar fram olika tal och skriver ut dem på skärmen. Flera olika funktioner finns beskrivna nedan och det är din uppgift att fylla dem med innehåll så att de utför angiven uppgift. Alla funktioner ska anropas ifrån main på ett sådant sätt att programmet utför det som står i listan nedan i den ordning som är angiven.

### Funktion: **kastaTarning**

Beskrivning: Simulerar ett tärningskast och skriver ut resultatet

Förvillkor: true

Eftervillkor: har skrivit ut ett slumpat tal mellan 1 och 6

### Funktion: **slumpaEntal**

Beskrivning: slumpar fram ett tal mellan 0 och 9 och skriver ut det

Förvillkor: true

Eftervillkor: har skrivit ut ett slumpat tal mellan 0 och 9

### Funktion: **singlaSlant**

Beskrivning: Simulerar att singla slant 3 gånger och skriver ut resultatet varje gång och det slutgiltiga resultatet

Förvillkor: true

Eftervillkor: har skrivit ut resultatet från tre omgångar med singla slant samt slutresultatet

### Funktion: **slumpaTal**

Beskrivning: Slumpar fram ett tal mellan low och high och skriver ut det

Förvillkor: low < high

Eftervillkor: har skrivit ut ett slumpat tal mellan low och high

## Main

1. Tärningskast
2. Slumptal mellan 0-9
3. Singla slant
4. Slumptal mellan 0-100
5. Slumptal mellan 5-25

### Del 3

1. Vad blir utskriften?

```
#include <iostream>
```

```
using std::cout;  
using std::endl;  
using std::cin;
```

```
void f(int a, int b)  
{  
    cout << a <<endl;  
    b = 2;  
    cout << b <<endl;  
}
```

```
void g(int b, int d)  
{  
    b = 5;  
    d = b + d;  
    cout << b <<endl;  
    cout << d <<endl;  
}
```

```
void h(int c, int d)  
{  
    d = c;  
    c = 4;  
    cout << d <<endl;  
    cout << c <<endl;  
}
```

```
void k(int a, int c)  
{  
    a = 3;  
    g(a, c);  
    cout << a <<endl;  
    cout << c <<endl;  
}
```

```
int main()  
{  
    int a = 5, b = 7, c = 3, d = 9;  
    cout << a <<endl;  
    cout << b <<endl;  
    cout << c <<endl;  
    cout << d <<endl;  
}
```

```
f(a, b);  
cout << a <<endl;  
a = 8;  
cout << b <<endl;  
g(b, d);  
cout << b <<endl;  
cout << d <<endl;  
h(c, d);  
cout << c <<endl;  
cout << d <<endl;  
k(a, b);  
cout << a <<endl;  
cout << b <<endl;  
cout << c <<endl;  
return 0;  
}
```

#### Del 4

1. Vilka utskrifter ger nedanstående program? Vilka är de formella respektive de aktuella parametrarna för funktionerna nedan ?

```
#include < iostream >  
using std::cout;  
using std::endl;  
  
int x, y, z;  
  
void a(int x)  
{  
    int y;  
    x = -2*x;  
    y = -2;  
    cout << x << y << z << endl;  
}  
  
void b(int &z)  
{  
    int x;  
    x = -2;  
    z = -2*z;  
    cout << x << y << z << endl;  
}  
  
int f(int a)  
{  
    y = a*a;  
    return y;  
}  
  
int main()
```

```
{
    x = 3; y = 4; z = 5;
    a(y);
    cout << x << y << z << endl;
    x = 3; y = 4; z = 5;
    b(x);
    cout << x << y << z << endl;
    x = 3; y = 4; z = f(x) + 2*y;
    cout << x << y << z << endl;

    return 0;
}
```

2. Lägg till en funktion i programmet som slumpar, del 2, som låter användaren mata in övre och under gräns. Anropa funktionen från main. Anropa därefter funktionen slumpaTal med de angivna gränserna. Glöm inte att kontrollera förvillkoret.

### Del 5

1. Titta på ditt program från del 2 där du slumpade fram olika saker i flera funktioner. Skapa en kopia av din cpp-fil som du kan ändra i. Skapa en h-fil och flytta över alla inkluderingar till denna. Lägg även in funktionsdeklarationer i h-filen. Inkludera nu h-filen i cpp-filen och kompilera och testkör. Det ska bli exakt samma resultat som tidigare.
2. Prova nu att byta plats på funktionerna i de bägge cpp-filerna (dels den gamla och dels den nya kopian). Flytta main och lägg längst upp i filen. Hur går det att kompilera de bägge filerna nu?

### Del 6

1. Skriv kod för följande funktioner. Skriv även anrop till dem från main som visar att de fungerar som beskrivet. Tips se kap 18.9

```
/*
```

**Avgör om en bokstav är versal eller gemen.**

*pre:* isalpha(bokstav)

*post:* har returnerat true om bokstav är versal och false om gemen

*parameter:* char bokstav

*returnerar:* bool true om versal och false om gemen

```
*/
```

```
/*
```

**Byter case på bokstaven.**

*pre:* isalpha(bokstav)

*post:* om bokstav var gemen är den nu versal och om bokstav var versal är den nu gemen

*parameter:* char bokstav

*returnerar:* -

```
*/
```

/\*

**Om tecken är en siffra får tal samma värde annars är tal -1.**

*pre:* true

*post:* om tecken är en siffra är tal = tecken  
annars är tal = -1

*parameter:* char tecken, int tal

*returnerar:* -

\*/

2. Nedan visas skelettet av ett simpelt orderprogram dvs funktioner med pseudokod inuti. Skriv för- och eftervillkor för funktionerna. På flera ställen skulle det även behövas kontroller. I tex funktionen nyOrderpost, efter att man skrivit in varunr, så skulle ju det behöva kontrolleras om det angivna varunr är giltigt eller inte. Lägg in kontroller på de ställen där du anser att det behövs.

//pre:

//post:

**int skapaOrder(int kundnr)**

{

    //skapa en tom order  
    //lägg in nästa lediga ordernr  
    //lägg in kundnr  
    //returnera ordernr

}

//pre:

//post:

**void nyOrderpost(int ordernr)**

{

    //fråga efter varunr  
    //fråga efter antal  
    //lägg in posten på ordern  
    //beräkna pris

}

//pre:

//post:

**bool giltigtKundnr(int kundnr)**

    //om giltigt kundnr  
    //returnera true  
    //annars  
    //returnera false

//pre:

//post:

**bool giltigtVarunr(int varunr)**

    //om giltigt varunr  
    //returnera true  
    //annars

```
//returnera false

//pre:
//post:
bool tillgång(int varunr, int antal)
    //om tillräckligt många av varan
    //returnera true
    //annars
    //returnera false

//pre:
//post:
float beräknaPris(int varunr, int antal)
    //pris = varupris x antal
    //returnera pris
```