

Programutveckling : LEU480

Idag: Programutveckling

Vad förväntas ni kunna

Kunna ta fram krav på vad ett program ska göra.

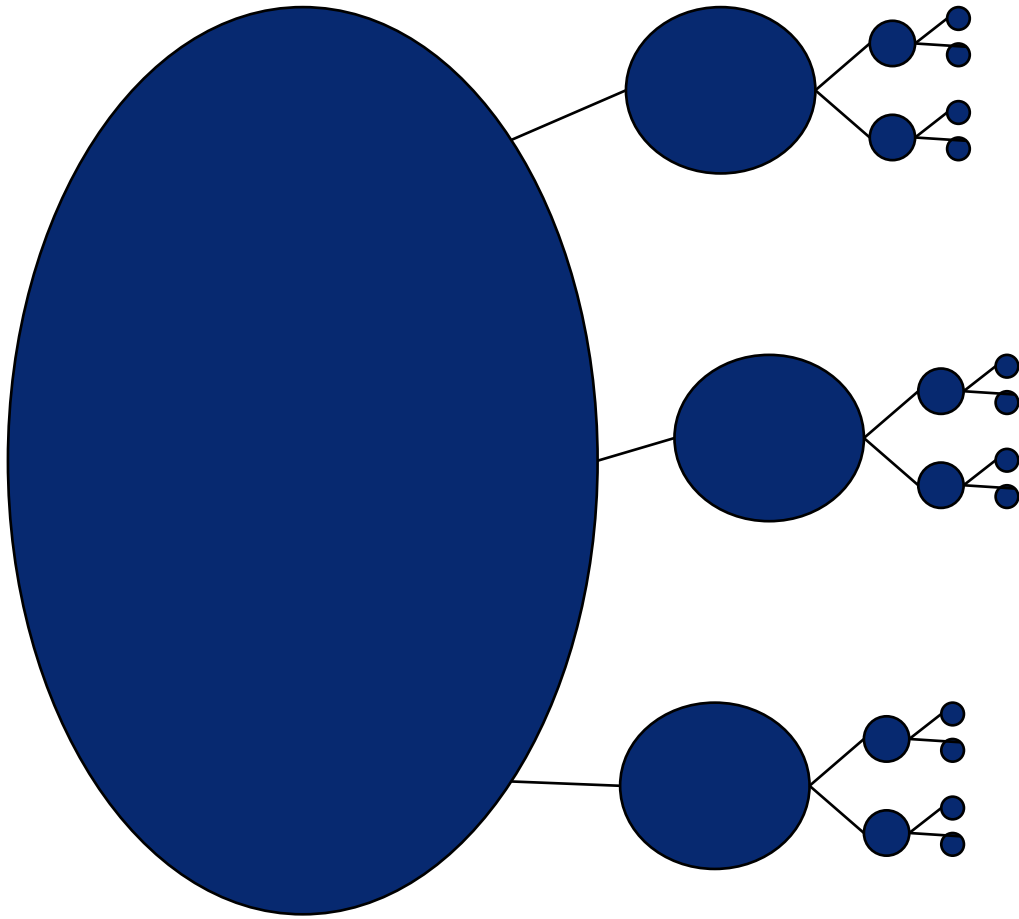
Kunna gruppera kraven utifrån vad som hör ihop med vad.

Kunna omsätta krav till ritade skärmbilder (ritat för hand eller i PowerPoint eller annan programvara).

Översätta ritade skärmbilder till kod.

Kunna fortsätta utveckla andra delar av ett program om du kört fast på en del av programmet.

Programutveckling – Från stort till litet



Programutveckling

Steg för steg

1. Lista på krav (vad ska programmet kunna åstadkomma/göra).
För att få underlag så fråga gärna användarna eller titta på hur de jobbar idag och gör noteringar utifrån det.
2. Försök gruppera kraven utifrån vilken ordning de ska sker och vilka som kan tänkas ske tillsammans
3. Översätt krav till vilka skärmbilder skall användaren mötas av.
(Menyer, inmatning av värden till formler, resultatutskrifter).
Rita gärna skärmbilder i PowerPoint eller på papper.
(En skärmbild representeras av att text skrivits ut på skärmen samt att man begär inmatning från användaren, mao inte hela programmet från start till slut).
4. Starta din utvecklingsmiljö och utifrån skärmbilderna, som ritats i punkt 3, skriv ner vilken C-kod som åstadkommer respektive händelse på skärmbilderna.

Iteration

Skapa först en program med minimal funktionalitet och testkör. Bygg sedan successivt ut programmet med mer funktionalitet.

Problemhantering

Medan du skriver C-kod kan du vid flera tillfällen känna att du inte vet hur du ska lösa en viss uppgift. Exempelvis beräkna lyftkraft. Skapa då en funktion som heter `beraknaLyftkraft()`; Du behöver inte i just detta ögonblick veta hur du ska göra den uppgiften utan anta bara att den på något sätt löser uppgiften och gå sedan vidare med utvecklingen av ditt program. Sätt även upp på din att-göra-lista att du behöver fylla funktionen `beraknaLyftkraft()`; med kod som löser uppgiften och lös den uppgiften vid ett senare tillfälle efter rådgivning från internet/kurslitteratur/arbetskamrater/klasskamrater/gruppmedlemmar/lärare/etc.

Programutveckling – ett praktiskt exempel

1.

Programmet tärningskast.

1. krav

• Två tärningar med nr: 1-6 (vanligtärning)
 Sätta pengar in
 Gissa utfallet vid tärningskast.
 Tärningssumman ^{2 & 12} ~~2 & 12~~ Ger 3x pengarna
 3 & 11 Ger 2,5x -"-
 4 & 10 Ger 2,0x -"-
 5 & 9 Ger 2x -"-
 6 & 8 Ger 2x Pengarna
 7 Ger 1,5x pengarna

- Vinst ges bara om man gissat rätt.
- programmet avslutas när man sätter in pengar
- Vinst utbetalas efter varje tärningskast om man vann
- När programmet avslutas skall man hälsas välkommen åter.

2.

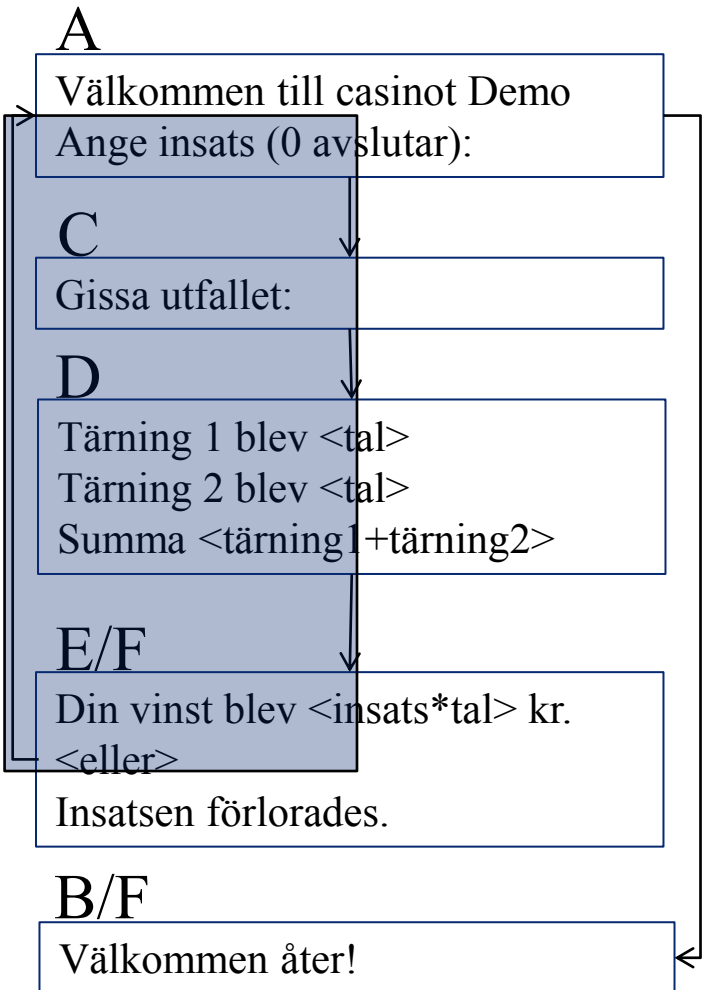
Programmet tärningskast.

1. krav

• Två tärningar med nr: 1-6 (vanligtärning)
 A Sätta pengar in
 C Gissa utfallet vid tärningskast.
 D Tärningssumman ^{2 & 12} ~~2 & 12~~ Ger 3x pengarna
 3 & 11 Ger 2,5x -"-
 4 & 10 Ger 2,0x -"-
 5 & 9 Ger 2x -"-
 6 & 8 Ger 2x Pengarna
 7 Ger 1,5x pengarna

- ~~A~~ • Vinst ges bara om man gissat rätt.
- B. programmet avslutas när man sätter in pengar
- ~~F~~ • Vinst utbetalas efter varje tärningskast (om man vann).
- G. När programmet avslutas skall man hälsas välkommen åter.

Programutveckling – ett praktiskt exempel – steg 3 & 4



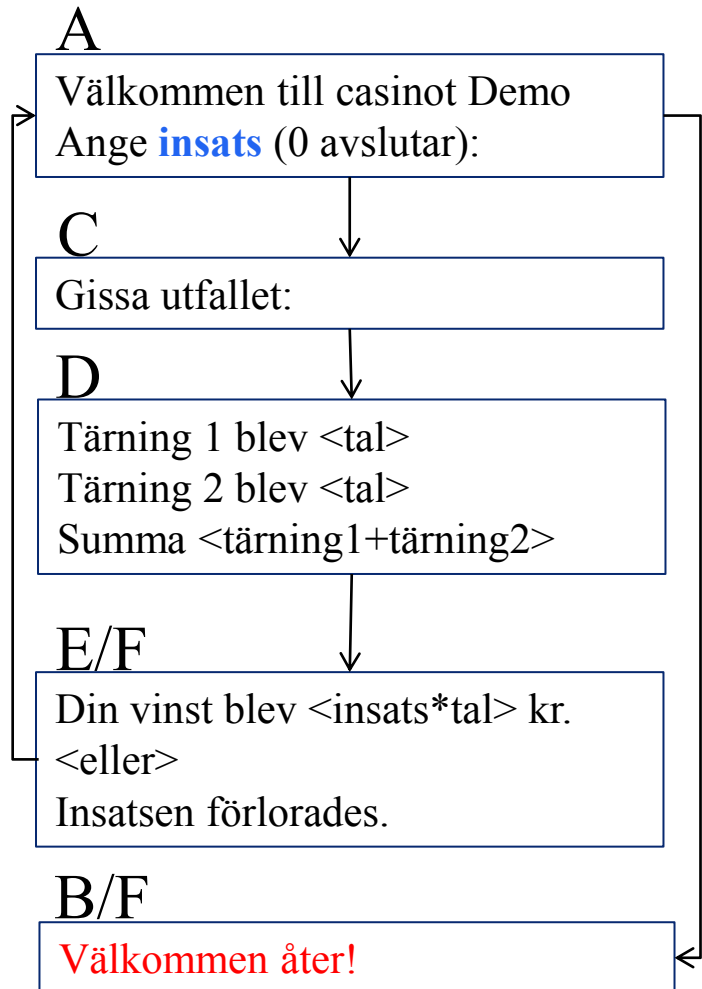
```
int main() {
```

```
    while (          ) {
```

```
    }
```

```
}
```

Programutveckling – ett praktiskt exempel – steg 3 & 4



```

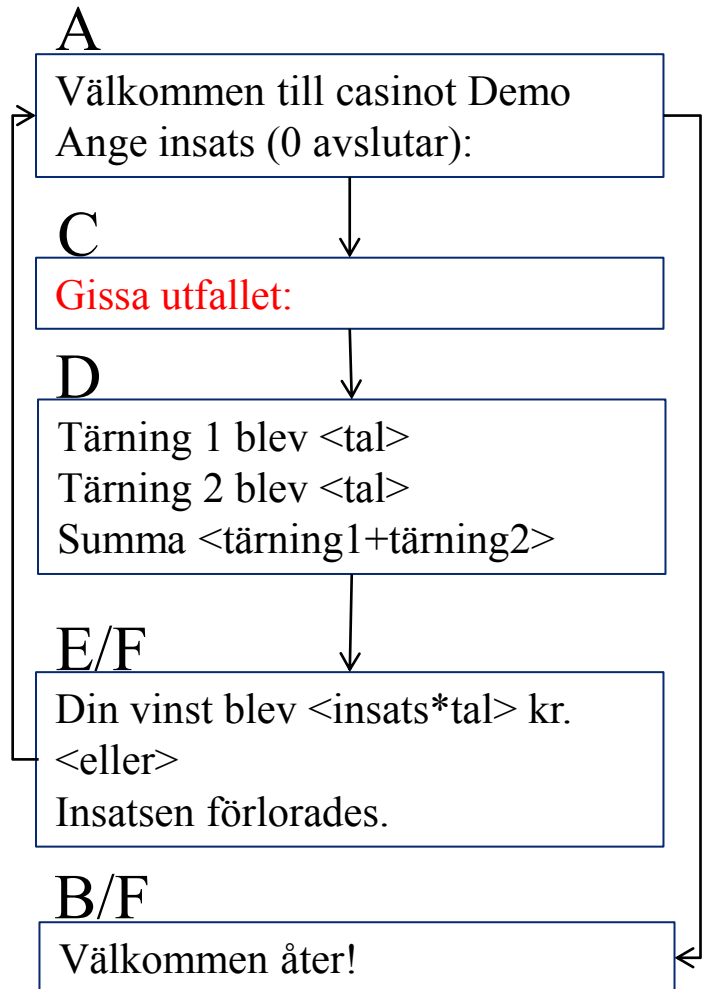
int main(){
    int insats = -1;

    printf("Välkommen till casinot Demo\nAnge insats (0 avslutar): ");
    scanf("%d", &insats);
    while (insats != 0) {

    }

    puts("\nVälkommen åter!");
}
  
```

Programutveckling – ett praktiskt exempel – steg 3 & 4



```

int main(){
    int insats = -1;

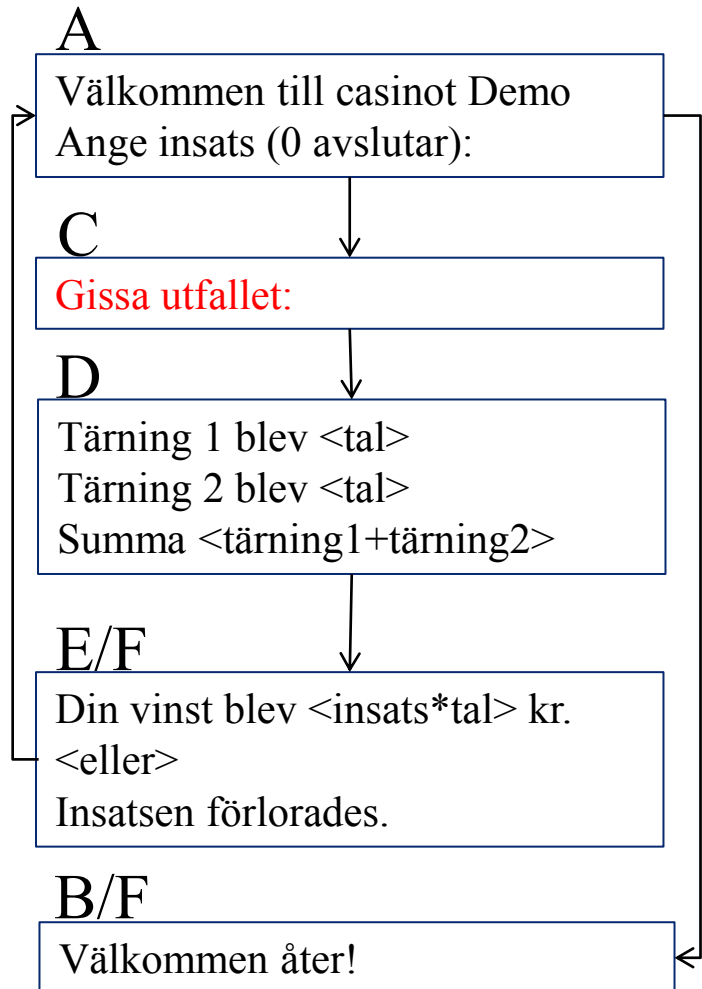
    printf("Välkommen till casinot Demo\nAnge insats (0 avslutar): ");
    scanf("%d", &insats);
    while (insats != 0) {
        printf("Gissa utfallet:");

        // ... (rest of the loop logic) ...

    }
    puts("\nVälkommen åter!");
}

```

Programutveckling – ett praktiskt exempel – steg 3 & 4



```

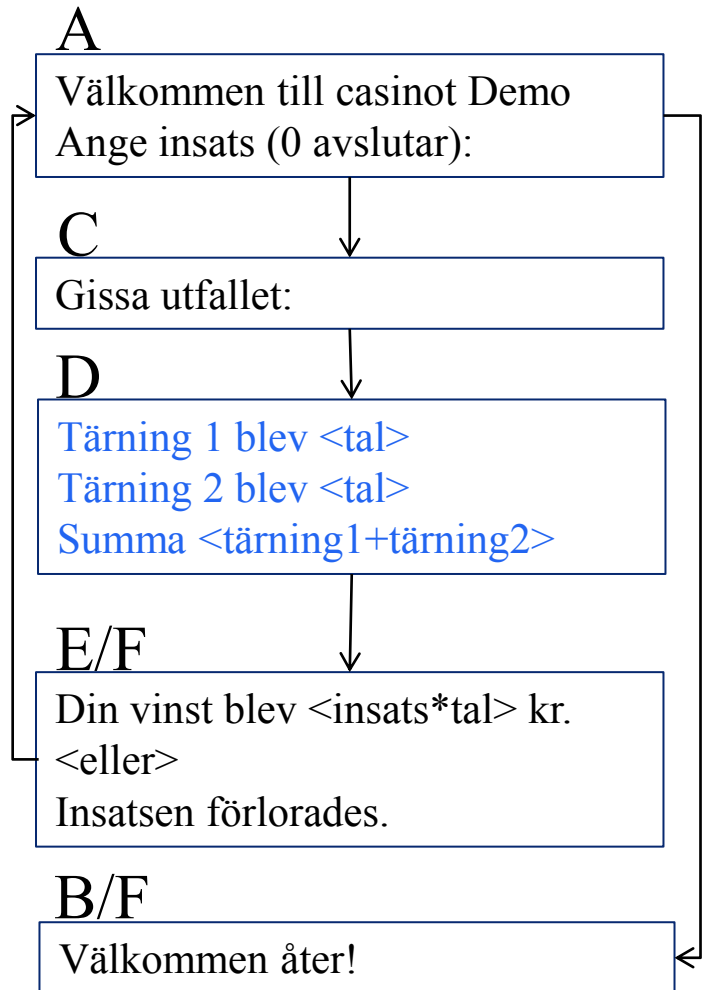
int main(){
    int insats = -1, gissning = 0;

    printf("Välkommen till casinot Demo\nAnge insats (0 avslutar): ");
    scanf("%d", &insats);
    while (insats != 0) {
        printf("Gissa utfallet:"); scanf("%d", &gissning);

        }
    puts("\nVälkommen åter!");
}

```


Programutveckling – ett praktiskt exempel – steg 3 & 4



```

#include <stdlib.h> // srand & rand
#include <time.h>   // time

int main(){
    int insats = -1, gissning = 0, dice1 = 0, dice2 = 0;

    srand(time(0)); // Generera slumpstal med utgångspunkt i vad klockan är
    printf("Välkommen till casinot Demo\nAnge insats (0 avslutar): ");
    scanf("%d", &insats);
    while (insats != 0) {
        printf("Gissa utfallet:"); scanf("%d", &gissning);
        dice1 = rand()%6 + 1; dice2 = rand()%6 + 1;
        printf("Tärning 1 blev %d\nTärning 2 blev %d\n", dice1, dice2);
        printf("Summa: %d\n", dice1 + dice2);

    }

    puts("\nVälkommen åter!");
}

```

Programutveckling – ett praktiskt exempel – steg 3 & 4

A

Välkommen till casinot Demo
Ange insats (0 avslutar):

C

Gissa utfallet:

D

Tärning 1 blev <tal>
Tärning 2 blev <tal>
Summa <tärning1+tärning2>

E/F

Din vinst blev <insats*tal> kr.
<eller>
Insatsen förlorades.

B/F

Välkommen åter!

```
#include <stdlib.h> // srand & rand
#include <time.h>    // time

int main(){
    int insats = -1, gissning = 0, dice1 = 0, dice2 = 0;
    float multiplikator = 0.0;
    srand(time(0)); // Generera slumpstal med utgångspunkt i vad klockan är
    printf("Välkommen till casinot Demo\nAnge insats (0 avslutar): ");
    scanf("%d", &insats);
    while (insats != 0) {
        printf("Gissa utfallet:"); scanf("%d", &gissning);
        dice1 = rand()%6 + 1; dice2 = rand()%6 + 1;
        printf("Tärning 1 blev %d\nTärning 2 blev %d\n", dice1, dice2);
        printf("Summa: %d\n", dice1 + dice2);

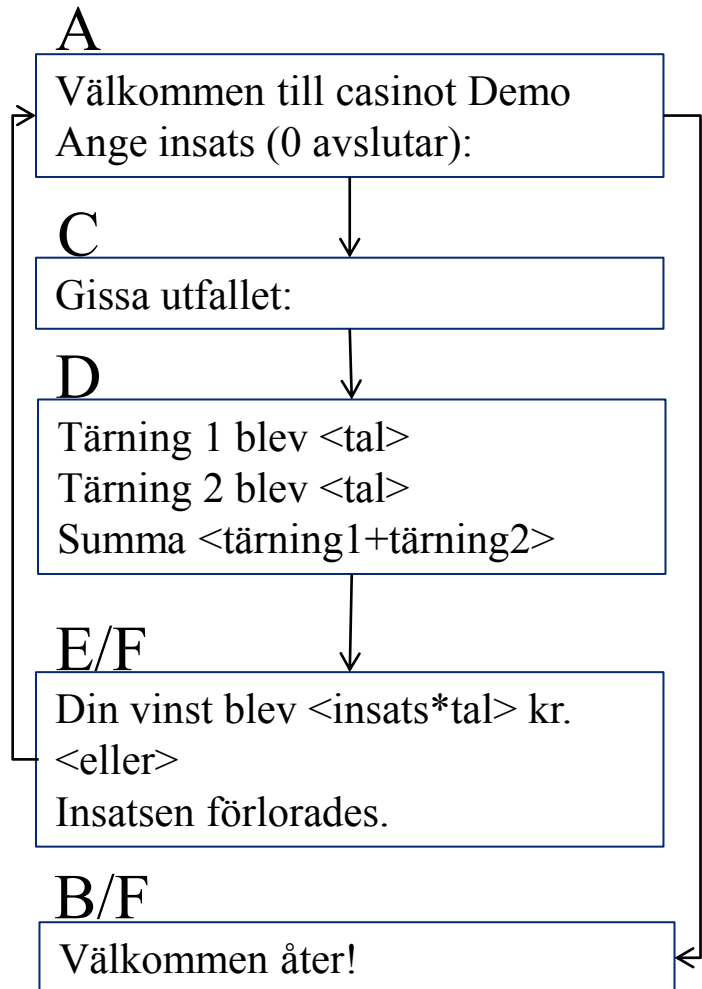
        if (gissning != (dice1 + dice2) ) { // Gissade spelaren fel?
            puts("Insatsen förlorades.");
        }
        else { // Annars vann spelare. Hur stor vinst ska utbetalas?
            if (gissning == 2 || gissning == 12) {
                multiplikator = 3.0;
            }
            else if (gissning == 3 || gissning == 11 ||
                    gissning == 4 || gissning == 10) {
                multiplikator = 2.0;
            }
            else if (gissning == 9 ||
                    gissning == 8) {
                multiplikator = 1.5;
            }
            else {
                multiplikator = 1.0;
            }
            printf("Din vinst blev %.2lf\n", (multiplikator * insats));
        }
    }

    puts("\nVälkommen åter!");
}
```

Tärnings Summa

2 & 12	Ger 3x pengarna
3 & 11	Ger 2,5x
4 & 10	Ger 2,0x
5 & 9	Ger 2x
6 & 8	Ger 2x pengarna
7	Ger 1,5x pengarna

Programutveckling – ett praktiskt exempel – steg 3 & 4



```

#include <stdlib.h> // srand & rand
#include <time.h>   // time

int main(){
    int insats = -1, gissning = 0, dice1 = 0, dice2 = 0;
    float multiplikator = 0.0;
    srand(time(0));
    printf("Välkommen till casinot Demo\nAnge insats (0 avslutar): ");
    scanf("%d", &insats);
    while (insats != 0) {
        printf("Gissa utfallet:"); scanf("%d", &gissning);
        dice1 = rand()%6 + 1; dice2 = rand()%6 + 1;
        printf("Tärning 1 blev %d\nTärning 2 blev %d\n", dice1, dice2);
        printf("Summa: %d\n", dice1 + dice2);

        if (gissning != (dice1 + dice2) ) {
            puts("Insatsen förlorades.");
        }
        else {
            if (gissning == 2 || gissning == 12) {
                multiplikator = 3.0;
            }
            else if (gissning == 3 || gissning == 11 ||
                     gissning == 4 || gissning == 10) {
                multiplikator = 2.5;
            }
            else if (gissning == 5 || gissning == 9 ||
                     gissning == 6 || gissning == 8) {
                multiplikator = 2.0;
            }
            else {
                multiplikator = 1.5;
            }
            printf("Din vinst blev %.2lf\n", (multiplikator * insats));
        }
        printf("Ange insats (0 avslutar): "); scanf("%d", &insats);
    } // Slut while
    puts("\nVälkommen åter!");
}
  
```

Vad förväntas ni kunna

Kunna ta fram krav på vad ett program ska göra.

Kunna gruppera kraven utifrån vad som hör ihop med vad.

Kunna omsätta krav till ritade skärmbilder (ritat för hand eller i PowerPoint eller annan programvara).

Översätta ritade skärmbilder till kod.

Kunna fortsätta utveckla andra delar av ett program om du kört fast på en del av programmet.