

Vad förväntas ni kunna

Förstå syftet med funktionsdeklaration.

Förstå vad typedef åstadkommer.

Kunna använda typedef.

Funktioners ordning

```
#include <stdlib.h> // srand & rand
#include <time.h>    // time
float multiplier(int gissning) {
    if (gissning == 2 || gissning == 12) { return 3.0; }
    else if (gissning < 5 || gissning > 9) { return 2.5; }
    else if (gissning < 7 || gissning > 7) { return 2.0; }
    else { return 1.5; }
}

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 {
            multiplikator = multiplier(gissning);
            printf("Din vinst blev %.2lf\n", (multiplikator * insats));
        }
        printf("Ange insats (0 avslutar): "); scanf("%d", &insats);
    } // Slut while
    puts("\nVälkommen åter!");
}
```

Funktioners ordning

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

```
float multiplier(int gissning); // Deklarera (berätta vad man har)
```

```
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 {
            multiplikator = multiplier(gissning);
            printf("Din vinst blev %.2lf\n", (multiplikator * insats));
        }
        printf("Ange insats (0 avslutar): "); scanf("%d", &insats);
    } // Slut while
    puts("\nVälkommen åter!");
}
```

```
float multiplier(int gissning) { // Definiera
    if (gissning == 2 || gissning == 12) { return 3.0; }
    else if (gissning < 5 || gissning > 9) { return 2.5; }
    else if (gissning < 7 || gissning > 7) { return 2.0; }
    else { return 1.5; }
}
```

Externa funktioner

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

```
float multiplier(int gissning);
```

```
int main() {
    int insats = -1;
    float multiplikator;
    printf("Välkommen till Chalmers!\n");
    scanf("%d", &insats);
    while (insats != 0) {
        printf("Gissning: ");
        int dice1 = rand() % 6 + 1;
        printf("Tärning 1: %d\n", dice1);
        printf("Summa: ");
        if (gissning == 2 || gissning == 12) {
            multiplikator = 3.0;
        } else if (gissning < 5 || gissning > 9) {
            multiplikator = 2.5;
        } else if (gissning < 7 || gissning > 7) {
            multiplikator = 2.0;
        } else {
            multiplikator = 1.5;
        }
        printf("Angiv gissning: ");
    } // Slut while
    puts("\nVälkommen till Chalmers!\n");
}
```

```
float multiplier(int gissning) {
    if (gissning == 2 || gissning == 12) {
        return 3.0;
    } else if (gissning < 5 || gissning > 9) {
        return 2.5;
    } else if (gissning < 7 || gissning > 7) {
        return 2.0;
    } else {
        return 1.5;
    }
}
```

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

```
// Filen externa.c
```

```
float multiplier(int gissning) {
    if (gissning == 2 || gissning == 12) { return 3.0; }
    else if (gissning < 5 || gissning > 9) { return 2.5; }
    else if (gissning < 7 || gissning > 7) { return 2.0; }
    else { return 1.5; }
}
```

```
if (gissning != (dice1 + dice2) ) { puts("Insatsen förlorades."); }
else {
    multiplikator = multiplier(gissning);
}
```

```
// Filen externa.h
```

```
float multiplier(int gissning);
```

typedef 1

Med typedef kan vi göra tre saker:

1. Ge nya namn åt befintliga datatyper
2. Skapa ”nya” datatyper
3. Göra kod mer lättläst.
4. Skapa flervärdesvariabler.

1. Ge nya namn åt befintliga datatyper

Ni minns **long long int** som kan innehålla heltal från -9.223.372.036.854.775.807 till + 9.223.372.036.854.775.808.

Att skriva **long long int** kan vara jobbigt. Vi kan förkorta det: **typedef long long int storint;**

Alltså, istället för att skriva **long long int** kan man nu skriva **storint**.

```
int main() {
    long long int antal = 0;
    // .....
    system("PAUSE");
    return (dag);
}
```

```
typedef long long int storint;

int main() {
    storint antal = 0;
    // .....
    system("PAUSE");
    return (dag);
}
```

typedef 2

Med typedef kan vi göra tre saker:

1. Ge nya namn åt befintliga datatyper
2. Skapa ”nya” datatyper
3. Göra kod mer lättläst.
4. Skapa flervärdesvariabler.

2. Skapa ”nya” datatyper

Låt oss säga att vi skall lagra någots vikt kilogram.: **typedef unsigned short vikt_i_kg;**

Alltså, istället för att skriva **unsigned short** kan man nu skriva **vikt_i_kg**.

Vi vet därmed att vikten är i kg och inte i exempelvis troy pound, gram eller någon annan viktenhet.

```
int main() {  
    unsigned short maxLastVikt = 0;  
    // .....  
    system("PAUSE");  
    return (dag);  
}
```

```
typedef unsigned short vikt_i_kg;  
  
int main() {  
    vikt_i_kg maxLastVikt = 0;  
    // .....  
    system("PAUSE");  
    return (dag);  
}
```

typedef 3

Med typedef kan vi göra tre saker:

1. Ge nya namn åt befintliga datatyper
2. Skapa ”nya” datatyper
3. Göra kod mer lättläst.
4. Skapa flervärdesvariabler.

3. Göra kod mer lättläst.

Istället för att tilldela värden så som -1, 0, 1 så kan vi tilldela värden så som ord.

```
#include <stdbool.h> // _Bool, true, false

typedef enum riktningar {DOWN=-1, STOPPED=0, UP=1} RIKT;
typedef enum vaningar {BASEMENT=-1, ENTRENCE=0, FIRSTFLOOR=1, ATTIC} van;

int main() {
    enum riktningar riktning = STOPPED;
    van nuvarandeVaning = ENTRENCE;
    enum vaningar onskadVaning = ENTRENCE;

    _Bool powerSupplyIsOn = true;

    ///////////////////////////////////
```

Vad förväntas ni kunna

Förstå syftet med funktionsdeklaration.

Förstå vad typedef åstadkommer.

Kunna använda typedef.