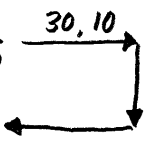


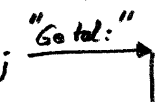
5) Funktioner

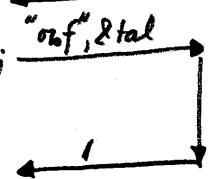
Anrop av biblioteksfunktioner

Ex) a) #include "MSViscon.h" //MSViscon.c i projektet

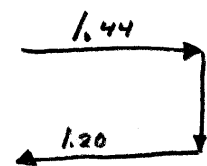
gotoxy(30,10);  gotoxy-funktion körs

b) #include <stdio.h>

printf("Ge tal: ");  printf körs

scanf("%f", &tal);  scanf körs

c) #include <math.h>

tal = sqrt(1.44);  sqrt körs

Skriver egna funktioner

- Ⓔ Funktion som skriver ett reellt tal i önskad position på skärmen.

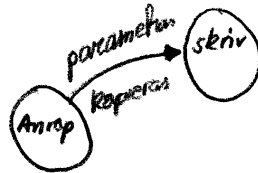
```
#include "MSViscon.h"
```

```
#include <stdio.h>
```

```
void skriv (float tal, int kol, int rad)
{
    gotoxy (kol, rad);
    printf ("%f", tal);
}
```

```
void main()
```

```
{
    skriv (5.6, 20, 10);
    skriv (1.3, 25, 12);
    |
}
```



Aktuella
parametervärden
kopieras till i
samma position
stående formella
parameter.

Information från anrop till funktion sker genom att aktuella parametervärden kopieras till motsvarande formella parameter.

Ex) Funktion som läser reellt tal från önskad position på skärmen om information från funktion till anrop sänds med

a) global variabel

```
#include <stdio.h>
#include "MSViscon.h"
```

```
float g; // Global variabel
```

```
void Las(int kol, int rad)
{
    gotoxy(kol, rad);
    printf("Ge reellt tal: ");
    scanf("%f", &g);
}
```

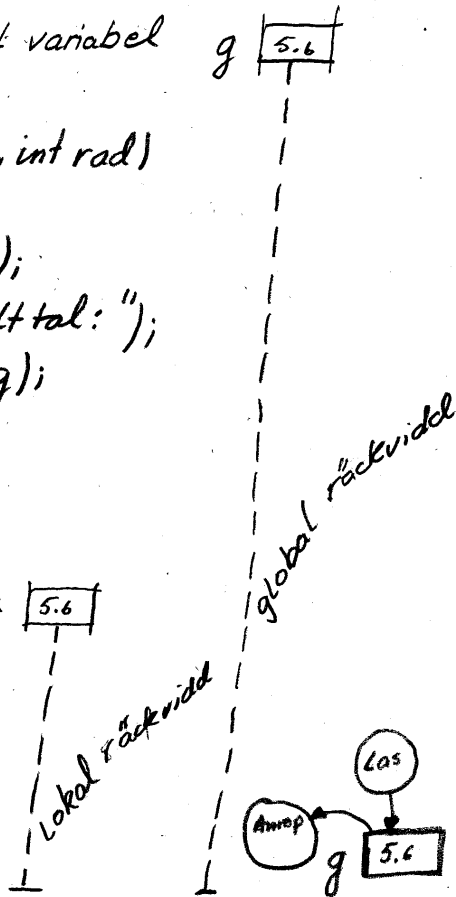
```
void main()
{
```

```
    float x; // Lokal x
              // variabel
```

```
    Las(30, 10);
```

```
    x = g;
```

```
}
```



Ej bra! Svårsumma fel! Ej generellt!

③

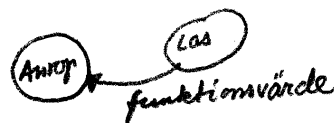
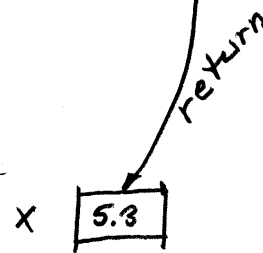
b) funktions värde

```
#include <stdio.h>
#include "MSViscon.h"

float las(int kol, int rad)
```

```
{
    float tal; // Lokal variabel tal [5.3]
    gotoxy(kol, rad);
    printf("Ge reellt tal: ");
    scanf("%f", &tal);
    return tal;
}
```

```
void main()
{
    float x;
    x = las(30, 12);
    ...
}
```



Funktionsvärdet är ofta upptaget av annan information ex. felkod och kan ej användas till att skicka information till anropet

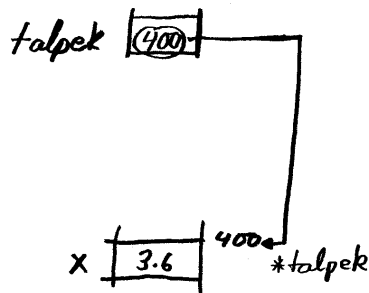
c) adressvariabel

```
#include <stdio.h>
#include "MSVIsLm.h"

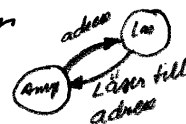
void las(float *talpek, int kol, int rad)
{
    gotoxy(kol, rad);
    printf("Ge reellt tal : ");
    scanf("%f", talpek); // OBS! Inget & ty
                          // & *talpek är talpek
}
```

```
void main()
{
    float x;

    Las(&x, 30, 12); // OBS! &x
    ...
}
```



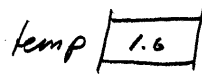
Skickar adress som kopieras till en pekare med vars hjälp man läser in talet till rätt ställe i minnet.



(5)

Ex) Funktion som byter värden mellan två reella tal.

```
void byt(float *xpek, float *ypek)
{
    float temp;
    temp = *xpek;
    *xpek = *ypek;
    *ypek = temp;
}
```



```
void main()
{
```

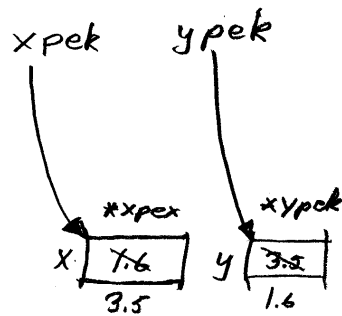
```
    float x=1.6, y=3.5;
```

```
    byt(&x, &y);
```

```
    printf("x = %.1f\n", x); // x=3.5
```

```
    printf("y = %.1f\n", y); // y=1.6
```

```
}
```



6

Ex) Funktion som tar ett heltal som parameter och returnerar sant (1) om talet är udda annars falskt (0).

```
int udda(int tal)
{
    if (tal % 2 == 0)
    {
        return 0;
    }
    return 1; // OBS! Behöver ej else
} // eftersom return alltid ger ett
// hopp ut ur funktionen
```

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int nr;
    printf("Ge ett heltal:");
    scanf("%d", &nr);
    if (udda(nr)) // OBS! Samma som udda(nr) == 1
        printf("%d är udda!", nr);
    else
        printf("%d är jämnt!", nr);
    getch();
}
```

(7)

Ex) Funktion som beräknar medelvärdet av en reell vektor.

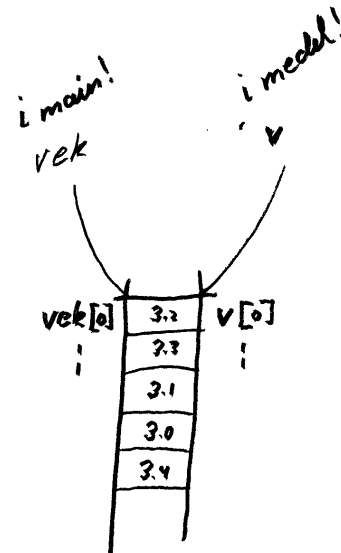
```
float medel(float v[], int nr) // Att float *v
{
    // nr är antalet element

    float sum = 0;
    int i;

    for(i=0; i<nr; i++)
    {
        sum += v[i];
    }

    if (nr > 0)
    {
        return sum/nr;
    }

    return 0;
}
```



```
#include <stdio.h>
#include <conio.h>
void main()
{
    float vek[5] = {3.2, 3.3, 3.1, 3.0, 3.4};
    printf("Medel = %.2f", medel(vek, 5));
    getch();
}
```

8

(Ex) Funktion som slumpar en heltalsvektor med angivet antal heltal mellan önskat minsta och största värde.

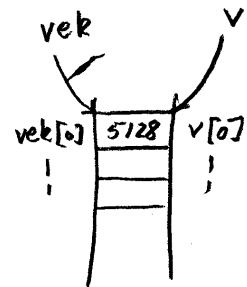
```
void slump(int v[], int nr, int low, int high)
```

```
{  
    int i, slumpstal;  
    srand((unsigned)time(NULL));  
    for (i=0; i<nr; i++)  
    {  
        slumpstal = rand()%(high-low+1)+low;  
        v[i]=slumpstal;  
    }  
}
```

```
#include <stdio.h>  
#include <conio.h>
```

```
void main()
```

```
{  
    int vek[100], i;  
    slump(vek, 100, 1000, 9999);  
    for (i=0; i<100; i++)  
        printf("%5d", vek[i]);  
    getch();  
}
```



- (Ex) a) Funktion som kontrollerar om ett tecken är en siffra (0-9).

```
int siffra(char ch)
{
    if (ch >= '0' && ch <= '9')
    {
        return 1;
    }
    return 0;
}
```

- b) Funktion som tar en sträng av siffror som parameter och returnerar motsvarande heltal. Ex "123" ska returneras som 123.

```
int tal(char *s) // Alternativ char s[]
{
    int sum = 0, i;
    for (i = 0; s[i] != '\0'; i++)
    {
        sum = sum * 10 + s[i] - '0';
    }
    return sum;
}
```

c) Program som läser en sträng och skriver det heltal som bildas av siffrorna i strängen

Körex

Ge en sträng: ab3c4kl1

Tal = 341

```
void main()
```

```
{
```

```
    char st[30], siffst[10];
```

```
    int i, k=0;
```

```
    printf("Ge en sträng: ");
```

```
    gets(st);
```

```
    for (i=0; st[i]!='\0'; i++)
```

```
    {
```

```
        if (siffa(st[i]) == 1)
```

```
        {
```

```
            siffst[k] = st[i];
```

```
            k++;
```

```
        }
```

```
    }
```

```
    siffst[k] = '\0';
```

```
    printf("Tal = %d", tal(siffst));
```

```
    |
```

```
}
```

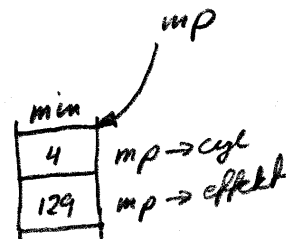
(EX) skriv funktioner för läsning och skrivning av

```
struct motor
{
    int cyl;
    float effekt;
};

void las_motor(struct motor *mp)
{
    printf("Ge motorns antal cylindrar : ");
    scanf("%d", &mp->cyl); // &(xmp).cyl
    printf("Ge motorns effekt : ");
    scanf("%f", &mp->effekt); // &(xmp).effekt
}

void skriv_motor(struct motor m)
{
    printf("Antal cylindrar : %d\n", m.cyl);
    printf("Effekt : %0.1f\n", m.effekt);
}

void main()
{
    struct motor min;
    las_motor(&min);
    skriv_motor(min);
}
```



Hemuppgift Dela upp lottosimuleringen med "Dracula" så att den har funktioner void skapa(int lottobollar[]) och void blanda(int lottobollar[], int antal_bollar)