

# Lecture 5 examples

November 19, 2018

## 1 Loops

and code repetition

### 1.1 goto label

- Provides a jump to from goto to a labeled statment
- **highly discouraged**
- and is a mark of poor programming skills

In [24]: `#include <stdio.h>`

```
int main()
{
    int i =0;
    start :
    printf ("%d " ,i );
    ++ i ;
    if (i <=5) goto start;
}
```

0 1 2 3 4 5

### 1.2 while (condition) {instructions}

first check the condition, than wxcute

In [25]: `#include <stdio.h>`

```
int main()
{
    int i=0;
    while(i < 10)
    {
        printf ("%d " ,i );
        ++ i ;
    }
}
```

0 1 2 3 4 5 6 7 8 9

### 1.3 do {instructions} while (condition)

Execute at least once, even if condition is not meet

In [28]: `#include <stdio.h>`

```
int main()
{
    int i=10;
    do
    {
        printf ("%d " ,i );
        ++ i ;
    }
    while(i < 10);
}
```

10

In [29]: `#include <stdio.h>`

```
int main()
{
    int i=0;
    do
    {
        printf ("%d " ,i );
        ++ i ;
    }
    while(i < 10);
}
```

0 1 2 3 4 5 6 7 8 9

### 1.4 for( initialization; condition; incrementation) {instructions}

- The initialization step is executed first, and only once
- Condition is checked before execution. Will not execute if initially condition is false!
- Incrementation is performed after the instructions are executed, as a last step
- initialization; condition; incrementation can be left out, as long as semicolons ; are in. I.e: for(;;) will work and result in an infinite loop

In [30]: `#include <stdio.h>`

```
int main()
{
    for(int i=0; i<10; ++i)
    {
        printf ("%d " ,i );
    }
}
```

0 1 2 3 4 5 6 7 8 9

Mind where variable is declared!

In [33]: `#include <stdio.h>`

```
int main()
{
    int j=0;
    for(int i=0; i<10; ++i)
    {
        printf ("i=%d " ,i );
        j = i;
    }
    printf (" | j=%d " ,j );
}
```

i=0 i=1 i=2 i=3 i=4 i=5 i=6 i=7 i=8 i=9 | j=9

An example of declaration before the loop

In [35]: `#include <stdio.h>`

```
int main()
{
    int i = 0;
    for(i=0; i<10; ++i) // but also for( ; i<10; ++i)
    {
        printf ("i=%d " ,i );
    }
    printf (" | I=%d " ,i );
}
```

i=0 i=1 i=2 i=3 i=4 i=5 i=6 i=7 i=8 i=9 | I=10

## 1.5 Nested loops

a for in a for

Write a program that prints a 4x5 (4 rows and 5 collumns) array.  $A_{ij} = i + j$

In [43]: `#include <stdio.h>`

```
int main()
{
    for(int i=0; i<4; ++i)
    {
        for(int j=0; j<5; ++j) // no brackets since only a single instruction
            printf ("%d\t" , i+j);
        printf("\n"); // new line
    }
}
```

0	1	2	3	4
1	2	3	4	5
2	3	4	5	6
3	4	5	6	7

## 1.6 Infinite loops

Sometimes (intentionally or not) an infinite loop is created (see one of the examples below) \*  
`while(1) {}` \* `for(;;) {}`

## 1.7 Flow control

i.e. how to manipulate execution of a loop, stop it, or skip the remaining instructions

### 1.7.1 break

The loop is terminated and the code following the loop is executed

Here a program that prints numbers, but stops as soon as the first number dividable by 7, that is not 7, is encountered.

In [46]: `#include <stdio.h>`

```
int main()
{
    for(int i=0; i<100; ++i)
    {
        printf("%d ", i);
        if (i%7 == 0 && i != 7 && i>0)
        {
            //printing ?
            break;
        }
    }
}
```

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14

### 1.7.2 continue

The loop execution is stooped, and started from the beginning

Similar as before, but this time program prints numbers, but skips those dividable by 7.

In [49]: `#include <stdio.h>`

```
int main()
{
    for(int i=0; i<30; ++i)
    {
```

```

        if (i%7 == 0)
        {
            //printing ?
            continue;
        }
        printf("%d ", i);
    }
}

```

1 2 3 4 5 6 8 9 10 11 12 13 15 16 17 18 19 20 22 23 24 25 26 27 29

## 1.8 Examples

### 1.8.1 Print odd even numbers

use continue

**This is an example from a test**

Write a complete program printing odd numbers from 0 to 50, but not those dividable by 9.  
The program

In [52]: `#include <stdio.h>`

```

int main()
{
    for(int i=1; i<50; i+=2)
    {
        if(i%9 != 0)
            printf("%d ", i);
    }
}

```

1 3 5 7 11 13 15 17 19 21 23 25 29 31 33 35 37 39 41 43 47 49

An alternative approach

In [53]: `#include <stdio.h>`

```

int main()
{
    for(int i=0; i<50; ++i)
    {
        if(i%2 == 0) continue;
        if(i%9 != 0)
            printf("%d ", i);
    }
}

```

1 3 5 7 11 13 15 17 19 21 23 25 29 31 33 35 37 39 41 43 47 49

## 1.8.2 Write a winning game

In [ ]: `#include <stdio.h>`

```
int main()
{
    while(1)
    {
        int uv = 0;
        printf("Give me a value that is not greater than 10\n");
        scanf("%d", &uv);
        int aiv = uv + 1;
        if(uv > 10)
        {
            printf("You cheat, I win!!\n");
            break;
        }
        printf("My is greater I win!!\n");
        printf("%d %d\n", uv, aiv);
    }
}
```

## 1.8.3 Fibonacci sequence

Write a program that prints the [Fibonacci numbers](#), those are numbers that belong to a Fibonacci sequence, i.e. each number in a sequence after the first two is sum of the two preceding ones

$$\begin{aligned}n_0 &= 0 \\n_1 &= 1 \\&\dots \\n_i &= n_{i-1} + n_{i-2}\end{aligned}$$

In [4]: `#include <stdio.h>`

```
int main()
{
    int n0 = 0;
    int n1 = 1;
    int nc = 0;
    for(int i=0; i<20; ++i)
    {
        nc = n0 + n1;
        n0 = n1;
        n1 = nc;
        printf("%d ", nc);
    }
}
```

1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765 10946

## 1.9 Fourier expansion of a square wave

Write a program, calculating the Fourier expansion of a square wave.

$$f(x) = \frac{4}{\pi} \sum_{n=1}^N \frac{a}{n} \sin\left(\frac{n\pi x}{L}\right)$$

```
In [ ]: %%cflags:-lm

#include <stdio.h>
#include <math.h>
#define PI 4.0*atan(1.0)

double fun(double x, int N, double L)
{
    double result = 0.0;
    for(int i=1; i<=N; ++i)
    {
        result += 1.0/i * sin((i*PI*x)/L);
    }
    return 4.0 / PI * result;
}

int main()
{
    int N = 2;
    double L = 2*PI;
    double x0 = 0, x1 = L;
    double h = (x1-x0)/99;

    for(int i=0; i<100; ++i)
    {
        double x = x0 + i*h;
        double y = fun(x, N, L);
        printf("x=%lf y=%lf\n", x, y);
    }
}
```