1.

Write a program which will print values of the hyperbolic cosinus. In order to do this, first write a function calculating values of the hyperbolic cosinus: $\cosh(x) = \frac{e^x + e^{-x}}{2}$. The function should have one argument: `double x` and value of calculated $\cosh(x)$ should be returned as the value of the function. Then inside the `main` function calculate the values of $\cosh(x)$ for 5, 50 and 500 values of x from the range $< -2, 2 >$. Use following code:

```
int n = 10;
double a = -2;
double b = 2;
double x;
double h;
h = ;                           // ? - find the correct step h
int k = 0;

for(x = a; x <= b; x += h)
{
  k++;
  printf("x = %12.5lf cosh(x) = %12.5lf\n", x, cosh(x));
}

printf("loop where done %d times\n", k);
```

Check how many times the loops were executed and how many times their should be. Check the last value of the variable `x`. Try this after changing type of variables to `float`.

2.

Modify the program such that loop will be executed in unambiguous way.

3.

Modify the program by storing the values of `x` and `cosh(x)` inside arrays. The arrays should be allocated dynamically depending on the value of N. For the dynamic allocation use:

```
float *tab = (float*) malloc(N * sizeof(float));
```

Remember to free allocated memory before exiting the program:

```
free(tab);
```

Create a graph of the $cosh(x)$ function inside Excel.

4.

Try the following code to see how to use functions inside a logical expressions:

```
#include <stdio.h>

int fun1()
{
  printf("calling function 1\n");
  return 1;
}

int fun2()
{
  printf("calling function 2\n");
  return -1;
}

int main()
{
  if (fun1() > 0 || fun2() > 0)
  {
    printf("one of the functions returned value > 0");
  }

  return 0;
}
```

See the results of the execution of the program after changing order of the functions inside the logical expression used in `if` statement, using `&&` instead of `||` and values returned by functions.

5.

Try the following code:

```
double x = 0.5, y;
int a = 0;

if (100)
  printf("sqrt(x) = %lf\n", sqrt(x));

if (-1)
  printf("sin(x) = %lf\n", sin(x));

if (a = -5)
{
  printf("fabs(a*x) = %lf", fabs(a*x));
  a = 0;
}

if (a == 0)
  printf("pow(x,3) = %lf", pow(x, 3.));
```

Did the program print all the expected information? Do you know how to write a comparison with == operator in such a way that the operator = (used by mistake) will be treated by compiler as an error?