

Tutorial

Programming & Data Structure: CS 11001

Section - 4/D

Department of Computer Science and
Engineering

I.I.T. Kharagpur

Spring Semester: 2013 - 2014 (23.01.2014)

Download

**Download the file tut230114.pdf from
Programming & Data Structures ... of**

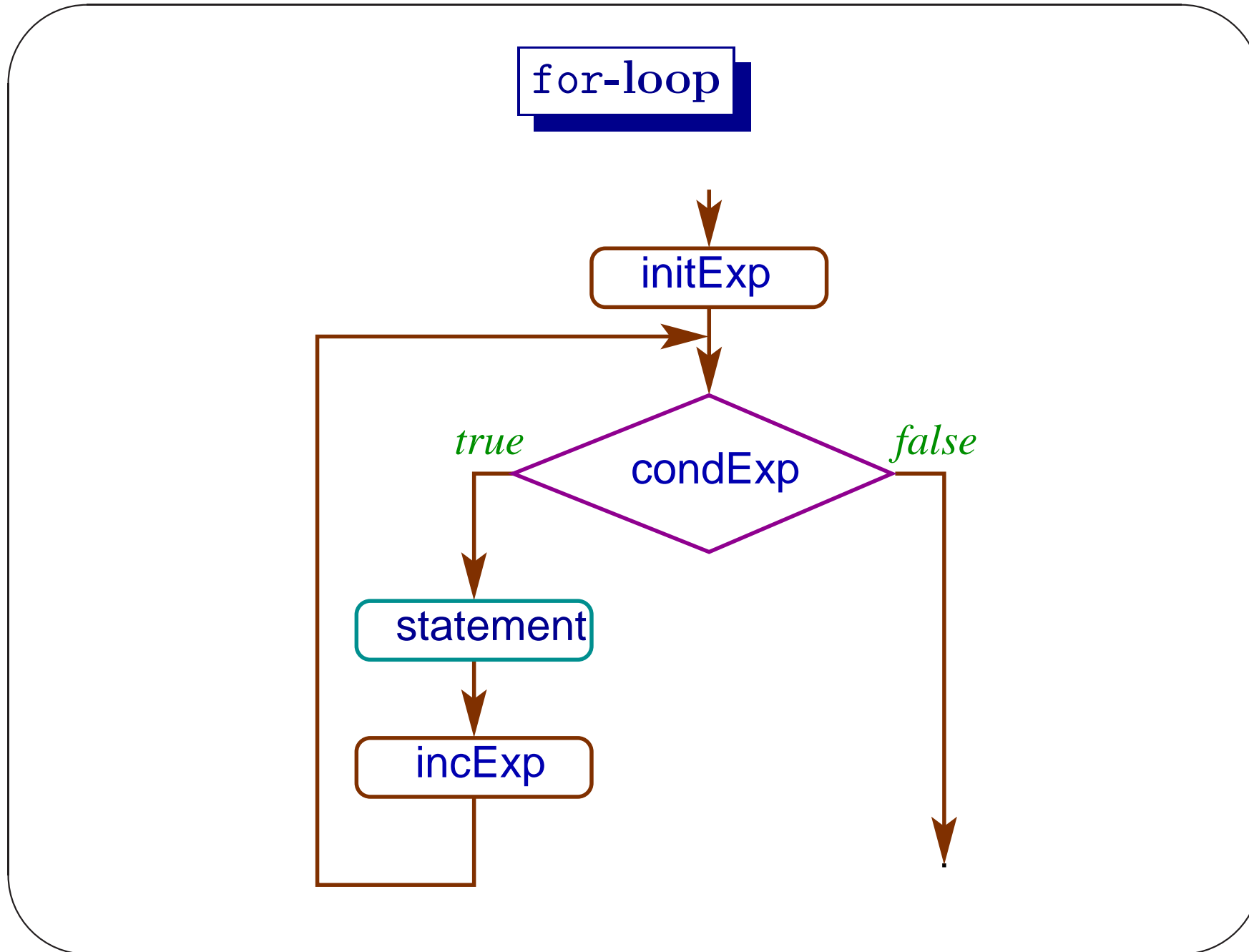
<http://cse.iitkgp.ac.in/~goutam>

**View the file using the command `acroread` & or
`xpdf` &**

for Statement

Another *iterative* construct in C language is the **for**-statement (loop). The structure or the **syntax** of this statement is,

```
for (initExp; condExp; incExp) statement
```



```
#include <stdio.h>
int main()          // forExp.c
{
    int n, i, sum, data;
    printf("Enter a +ve integer: ");
    scanf("%d", &n);
    printf("Enter %d data: ", n);
    for(i=1, sum=0; i<=n; ++i) {
        scanf("%d", &data);
        sum = sum + data;
    }
    printf("sum: %d\n", sum);
    return 0;
}
```

Tutorial III.1

Modify `forExp.c` to read a stream of n data and print the sum of those data in the stream that are not divisible by 3.

Input: 5

Input: 29 6 31 -52 27

output: 8

Tutorial III.2

Write a C program using **for-loop** to compute quadratic mean (QM) or root mean square (RMS) of n data. The only mathematical library function you are allowed to use is **sqrt(x)** that computes \sqrt{x} . Quadratic mean of n data is,

$$\sqrt{\frac{d_1^2 + \cdots + d_n^2}{n}}$$

Use **%f** format for input/output of **float** data.

Data

Input: 1

Input: 5

output: 5.0000

Input: 5

Input: 1 2 3 4 5

output: 3.316625

break Statement

A **break** statement forces the control out of the innermost **loop** or **switch** statement. It does not work for **if** statement.

```
break;
```

```
#include <stdio.h>
int main()          // breakExp.c
{
    float i, sum, err, term ;
    printf("Enter a small +ve value: ");
    scanf("%f", &err);
    for(i=2.0, sum=1.0; ; ++i) {
        term = 1.0/i;
        if(term < err) break;
        sum = sum + term;
    }
    printf("sum: %f\n", sum);
    return 0;
}
```

Tutorial III.3

Modify `breakExp.c` so that it computes an approximate sum of the series $\sum_{i=1}^{\infty} \frac{1}{i^2}$. The loop is terminated when the value of the next term $\frac{1}{i^2}$ is less than `err percent` of the current *approximate sum*. Also print the number of terms added.

Input: `err = 0.1`

Output: `sum: 1.604123, count: 24`

continue Statement

A **continue** statement forces the control immediately to the *test expression* of **while** and **do while**. It forces the control to *increment expression* in a **for** statement.

```
continue;
```

Tutorial III.4

Write a C program to compute and print the harmonic mean of a set of data using **for-loop** where the number of data is not known *a priori*. Assume that no data is equal to zero. If there are n data,

$$HM = \frac{n}{1/d_1 + 1/d_2 + \cdots + 1/d_n}.$$

Input: 1 2 3 4 5

Output: 2.189781

Tutorial III.5

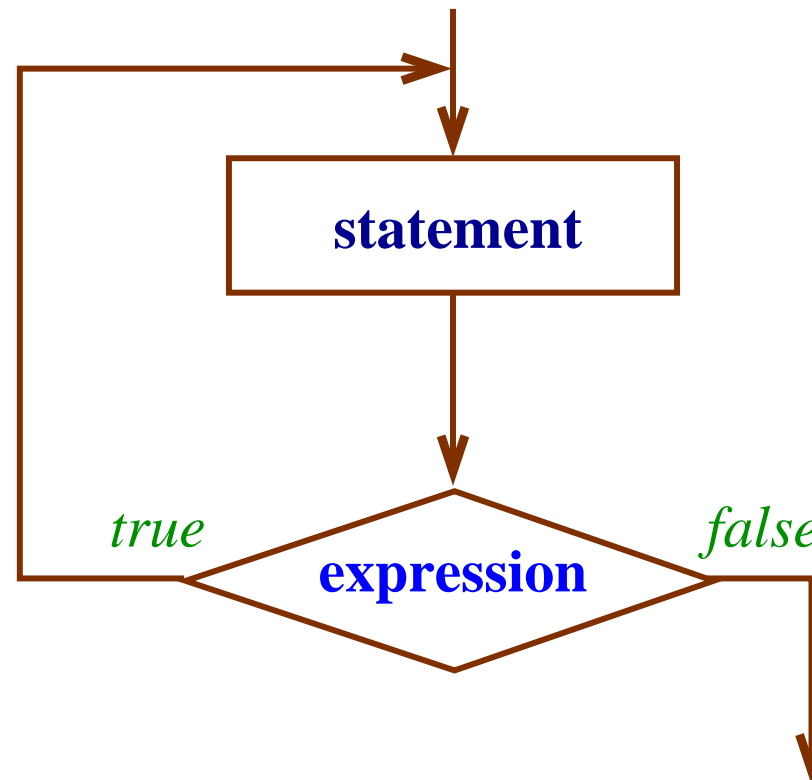
Read two positive integers s and l and print the $lcm(s, l)$ using `for-loop`.

do-while Statement

Another iterative construct of C language is

```
do statement while (expression) ;
```

do-while Loop



Tutorial III.6

Solve the problem of III.3 using **do-while** loop.