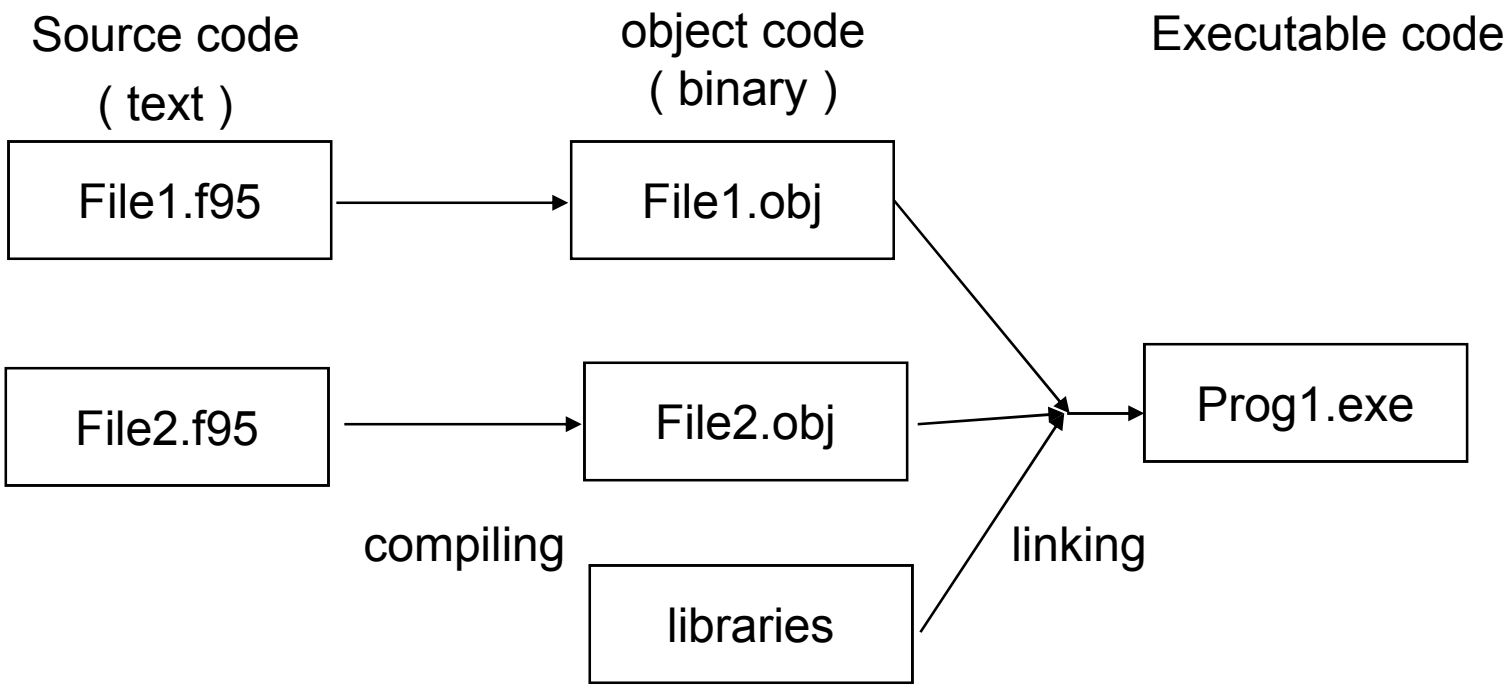
CITY UNIVERSITY LONDON

Creating Executable Code



- Source code: text file you write
- Executable code: binary code

CITY UNIVERSITY
LONDON

FUNCTIONS AND SUBROUTINES

All major computing languages allow complex and/or repetitive programs to be broken down into simpler procedures, each carrying out particular well-defined tasks, often with different values of certain parameters.

In Fortran these *subprograms are called* ***subroutines*** *and* ***functions****.*

Fortran95 Manual

9. FUNCTIONS AND SUBROUTINES
9.1 – 9.5

CITY UNIVERSITY LONDON

Intrinsic (build-in) Functions

Examples of intrinsic functions:

SQRT: y=sqrt(2.0)

ABS: y=abs(x)

SIN: y=sin(x) (x: radians)

ATAN: pi=4.0*ATAN(1.0)

When do we need user-defined functions?

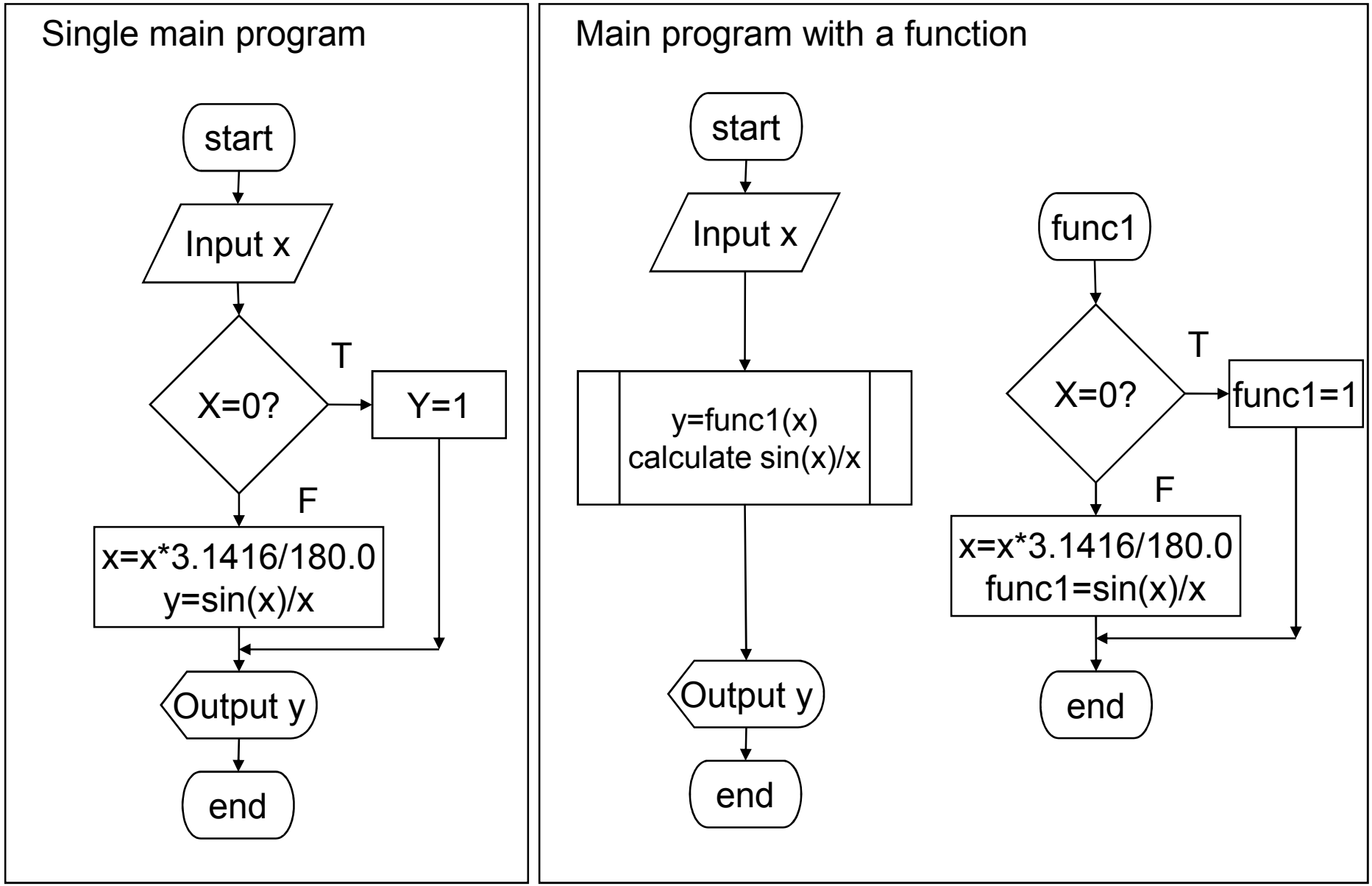
not in the intrinsic function list

long formulas

repeatedly used formulas

CITY UNIVERSITY LONDON

Program structure (function)



CITY UNIVERSITY LONDON

User-Defined Functions

General Form

[*type*] **FUNCTION** *name* (*argument-list*)
[IMPLICIT NONE]
type declarations
executable statements
END [FUNCTION [*name*]]

```
PROGRAM EXAMPLE
PRINT *, 'Input X, Y'
READ *, X, Y
r = RADIUS( X, Y )
PRINT *, 'Distance = ', r
END PROGRAM EXAMPLE
!=========================
REAL FUNCTION RADIUS( A, B )
RADIUS = SQRT( A ** 2 + B ** 2 )
PRINT *, A,B
END FUNCTION RADIUS
```

On with line is the function used?

Function name and type?

How many arguments in the function?

CITY UNIVERSITY LONDON

User-Defined Functions

Example

Write a function which calculates the mean value of 10 numbers stored in a real type array.

function name: mean (real type)
argument list: x (a real type array, 10 elements)

```
REAL FUNCTION mean(x)
real x(10), sum
sum=0
do i = 1, 10
   sum = sum + x(i)
end do
mean = sum/10
END FUNCTION mean
```

In Main program:

```
Real R(10), avg
Real mean          !external

Do i=1,10
  read(*,*) R(i)
End do

Avg = mean(R)
```

CITY UNIVERSITY LONDON

User-Defined Functions

Example

Write a function which calculates the mean value of N numbers stored in a real type array. A value is assigned to N in the main program.

function name: mean (real type)
argument list: x (a real type array, N elements), N

```
REAL FUNCTION mean(x, N)
integer N
real x(N), sum
sum=0
do i = 1, N
   sum = sum + x(i)
end do
mean = sum/N
END FUNCTION mean
```

In Main program:

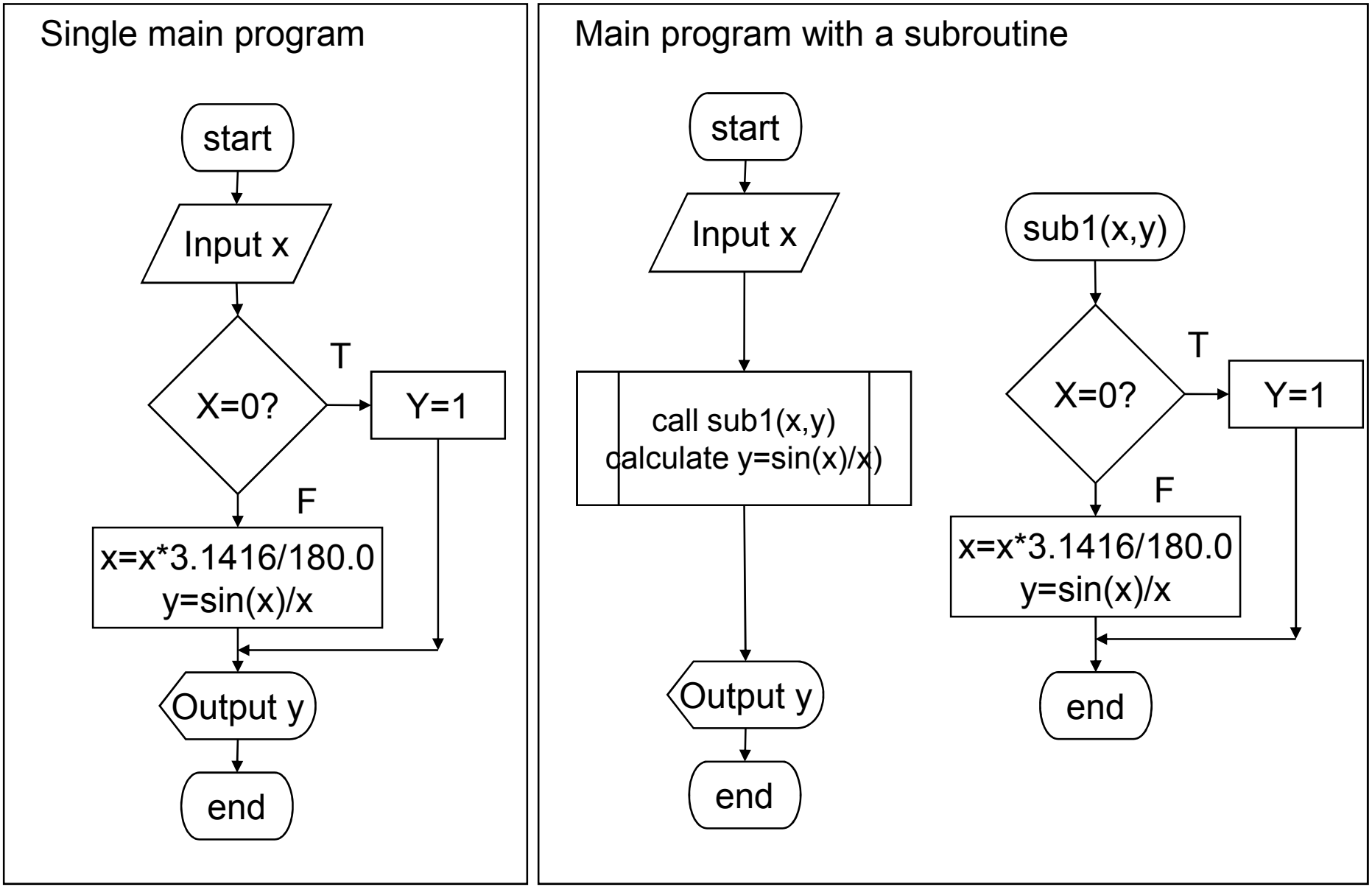
```
Real R(10), avg
Real mean            !external

Do i=1,N             ! N≤10
  read(*,*) R(i)
End do

Avg = mean(R, N)
```

CITY UNIVERSITY LONDON

Program structure (subroutine)



CITY UNIVERSITY LONDON

Subroutines

General Form

SUBROUTINE subroutine-*name* (*argument-list*)
[IMPLICIT NONE]
type declarations
executable statements
END [SUBROUTINE [*name*]]

Subroutines are self-contained

```
PROGRAM EXAMPLE
PRINT *, 'Input X, Y'
READ *, X, Y
CALL DISTANCE( X, Y, RADIUS )
PRINT *, 'Distance = ', RADIUS
END PROGRAM EXAMPLE
!=========================
SUBROUTINE DISTANCE( A, B, R )
R = SQRT( A ** 2 + B ** 2 )
END SUBROUTINE DISTANCE
```

On which line is the subroutine called?

subroutine name?

argument list?

intended data transfer direction?

What is the main difference compared with functions?

CITY UNIVERSITY LONDON

Subroutines

Example

Write a subroutine which calculates the mean value of N numbers stored in a real type array. A value is assigned to N in the main program.

subroutine name: get_mean
argument list: x (a real type array, N elements), N , mean

```
call get_mean(x, N, mean)
```

```
subroutine get_mean(x, N, mean)
integer N
real x(N), sum, mean
sum=0
do i = 1, N
   sum = sum + x(i)
end do
mean = sum/N
END subroutine get_mean
```

CITY UNIVERSITY LONDON

Subroutines

Exercise

Write a subroutine which calculates the mean value and the standard deviation of N numbers stored in a real type array. A value is assigned to N in the main program.

subroutine name: get_mean_std

argument list: x (a real type array, N elements), N , mean, std

call get_mean_std(x, N, mean, std)

```
subroutine get_mean_std(x, N, mean, std)
integer N
real x(N), sum, mean, std
sum=0
do i = 1, N
   sum = sum + x(i)
end do
mean = sum/N
( write the code for calculating std)
END subroutine get_mean_std
```

$$std = \sqrt{\frac{\sum_{i=1}^{N}(x(i) - mean)^2}{N}}$$

CITY UNIVERSITY LONDON

The standard deviation

The standard deviation:

$$std = \sqrt{\frac{\sum_{i=1}^{N}(x(i) - mean)^2}{N}}$$

Flowchart: