

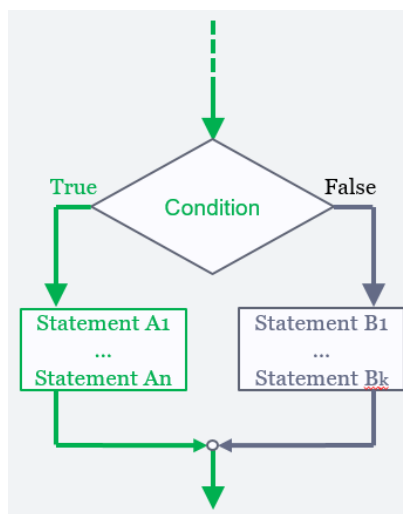
TPOP PRACTICAL

SELECTION

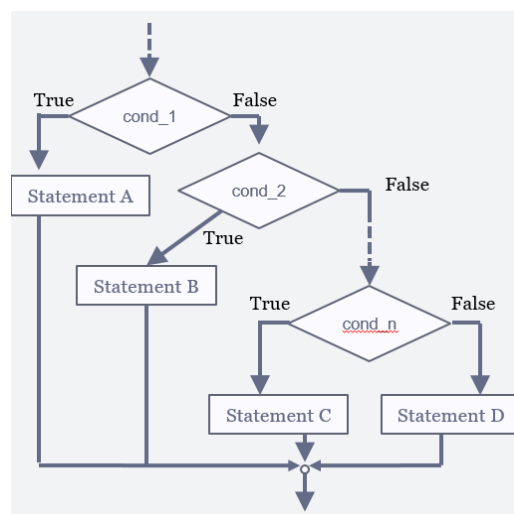
PRACTICAL 02

Testing is a key part of the software development process. You should think about it from day one, e.g. your first programs (they will be the easiest one to test). For each of the question below, write down the series of tests (values used) you will be using to validate your code. Are you sure you covered all cases? Compare your tests to someone else, are they equivalent?

In addition, this week practical will continue to focus on the selection statement `if-elif-else` seen during last week practical. The statement can be described by flow chart diagram:



if-else flow chart



if-elif-else flow chart

and below is their coding equivalents.

```

...
if condition :
    Statement A1
    ...
    Statement An
else :
    Statement B1
    ...
    Statement Bk
...
  
```

```

...
if cond_1 :
    Statement A
elif cond_2 :
    Statement B
...
elif cond_n :
    Statement C
else :
    Statement D
...
  
```

Note that the `else` statement is optional for both flow charts.

Solve questions 1 and 2 using the selection statement.

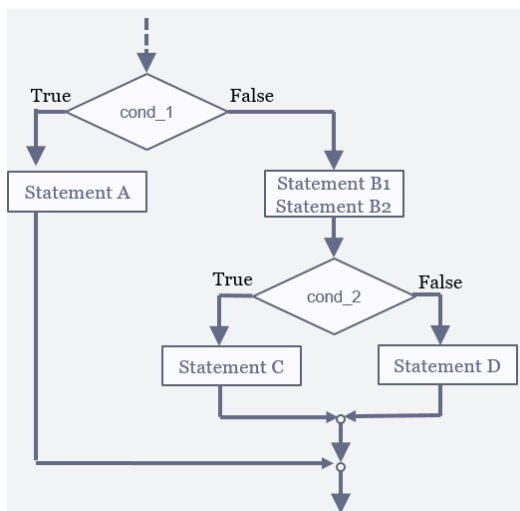
Question 1:

Write a program taking three numbers from the user and print the greatest values. Write a series of test you would use to check the correctness of your program.

Question 2: *Not applicable to students!*

Many companies pay time-and-a-half for any hours worked above the 40 in a given week. Write a program to input the number of hours worked in a week and the hourly rate and calculate the total wages for the week. It should also indicate the wages paid in overtime.

Sometime it could be useful to use nested if-elif-else statement, e.g. use another if-elif-else statement inside another if-elif-else statement. An example of such nested statement is given below with the accompanying flow chart:



```
...
if cond_1 :
    Statement A
else :
    Statement B1
    Statement B2
    if cond_2 :
        Statement C
    else :
        Statement D
...
```

Question 3: *You've been fined!*

The speeding ticket fine policy in Blotsville is £100 plus £5 for each mph over the limit plus a penalty of £200 for any speed over 90 mph. Write a program that accepts a speed limit and a clocked speed and either prints a message indicating the speed was legal or prints the amount of the fine, if the speed is illegal.

So far we have been using very simple conditional expressions, such as `number_cakes < 1`. We can use comparison operators such as `==` (equal), `!=` (not equal), `<` (strictly less than), `>` (strictly greater than), `<=` (less or equal to), and `>=` (greater or equal to) to create a Boolean expression, i.e. an expression that returns only True or False. Sometime we need to build more complex Boolean expression, for example check if the number of cakes is less than 2 or greater or equal to 10. Python provides three Boolean operators: `and`, `or`, `not` to combine Boolean expressions. Therefore, the expression “if the number of cakes is less than 2 or greater or equal to 10” can be written as:

```
if number_cakes < 2 or number_cakes >= 10:
```

It should be noted that there is more than one way to write that expression, for example

```
if not (2 <= number_cakes < 10):
```

You may need to write more complex Boolean expression to solve the next question. Remember to solve the problem on paper using plain English before attempting to write the code.

Question 4: *Having a date?*

1. A year is a leap year if it is divisible by 4, unless it is a century year that is not divisible by 400. For example 1800 and 1900 are not leap year while 1600 and 2000 are. Write a program that calculates whether a year is a leap year or not
2. Write a program that accepts a date in the format DD/MM/YYYY and output whether or not the date is valid. For example 20/07/1969 is valid, but 31/09/2012 is not.

Question 5: *Functions*

Rewrite questions 1-4 using functions. Do you need parameters, return value(s)?