**Сложные условия**

Предположим, что ООО «Кнут и Пряник» набирает сотруд-ников, возраст которых от 25 до 40 лет включительно. Нужно написать программу, которая запрашивает возраст претендента и выдает ответ: «подходит» он или «не подходит» по этому при-знаку.

Какое же условие должно быть истинно для того, чтобы че-ловека приняли на работу? Одного условия «*возраст* >= 25» не хватает, это условие соблюдается и для людей старше 40 лет. С другой стороны, условия «*возраст* <= 40» тоже не хватает, так как оно выполняется и для школьников. В этой задаче нужно, чтобы два **условия выполнялись одновременно**: «*возраст* >= 25» и «*возраст* <= 40».

*Пусть в переменной* **age** *записан возраст сотрудника. Запишите решение задачи с помощью вложенных условных операторов.*

Эту задачу можно решить с помощью вложенного условного оператора, но решение получается некрасивое: оно запутанное и, кроме того, один и тот же ответ «*не подходит*» приходится вы-водить в двух местах программы.

Почти во всех языках программирования в условном опе-раторе можно использовать такое условие:

**if age >= 25 and age <= 40:**

**print( "подходит" )**

**else:**

**print( "не подходит" )**

Решение получилось короткое и понятное. В условном операто-ре мы записали **сложное условие**

**age >= 25 and age <= 40**,составленное из двух простых с помощью логической операции «И». В языке Python эта операция обозначается словом **and**.

**Операция «И» (and) означает одновременное выполнение двух или нескольких условий.**

В программа на языке Python можно сразу проверить вы-полнение двойного неравенства:

**if 25 <= age <= 40:**

**print("подходит")**

**else:**

**print("не подходит")**

но во многих популярных языках программирования такая за-пись приводит к ошибке.

*Что будет выведено на экран после выполнения следующей программы?*

**if a == 1 and a == 2:**

**print( "Да!" )**

**else:**

**print( "Нет." )**

Предположим, что нам надо убедиться, что значение целой переменной *a* – трёхзначное число, которое делится на 7. Для этого нужно, чтобы одновременно выполнились три условия:

1) число не меньше 100;

2) число меньше 1000;

3) число делится на 7, то есть остаток от его деления на 7 равен нулю.

В условном операторе эти три простых условия должны быть связаны с помощью двух операций «И»:

**if 100 <= a and a < 1000 and a % 7 == 0:**

**print( "Да!" )**

**else:**

**print( "Нет." )**

Рассмотрим ещё одну задачу. Самолёт из Санкт-Петербурга в Барнаул летает только по понедельникам и четвергам. В пе-ременной *day* записан номер дня недели (1 – понедельник, 7 – воскресенье). Программа должна определить, полетит ли само-лёт в этот день.

Если мы напишем условие «**day** = 1 **and day** = 4», то это бу-дет неверно, потому что мы потребовали, чтобы значение пере-менной *day* было *одновременно* равно и 1, и 4. Такого быть не может, поэтому это условие всегда будет ложно. Значит, опера-ция «И» не подходит. Вместо неё нужно применить другую опе-рацию – «ИЛИ», которая требует выполнения хотя бы одного из набора условий.

**Операция «ИЛИ» означает выполнение хотя бы одного из двух или нескольких условий.**

Решение нашей задачи выглядит так:

**if day == 1 or day == 4:**

**print( "Полетит!" )**

**else:**

**print( "Нет рейса." )**

В языке Python операция «ИЛИ» обозначается словом «**or**» (по-английски – «или»).

Существует ещё одна операция, которую можно использо-вать в сложных условиях – «НЕ», в Python она обозначается словом «**not**» (по-английски – «не»).

**Операция «НЕ» означает обратное условие (противоположное исходному).**

Если исходное условие истинно, то обратное (противоположное) ему – ложно, и наоборот.

Например, решение задачи с самолётом можно было запи-сать так:

**if not( day == 1 or day == 4 ):**

**print( "Нет рейса." )**

**else:**

**print( "Полетит!" )**

Используя операцию «НЕ», можно записывать условия по-разному, как нам удобнее в каждом случае. Например, условия **a==b** и **not(a!=b)** истинны для одних и тех же значений **a** и **b**, поэтому одно из них можно заменить на другое. Такие условия называются *равносильными*.

*Запишите равносильные условия, не используя операцию «НЕ»:*

*а)* **not (a < 6)**

*б)* **not (b == c+d)**

*в)* **not (c != 15)**

**Порядок выполнения операций**

Операции «И», «ИЛИ» и «НЕ» – это **логические опера-ции**, то есть они применяются к логическим значениям: «да»/«нет», «истина»/«ложь».

Если в сложном условии встречается несколько разных операций, они выполняются в следующем порядке (во всех слу-чаях – слева направо):

1) операции в скобках;

2) операции «НЕ»;

3) операции «И»;

4) операции «ИЛИ».

Изменить порядок действий можно с помощью круглых скобок.

*Определите порядок операций при определении истинности условия:*

**not (a > 10) or not (a < 10) and (a < b)**

*Определите, истинно или ложно это выражения при a* = 5, *b* = 10*.*

**Домашняя работа:**

1. Напишите программу, которая определяет, принадлежит ли число *x* отрезку [*a*; *b*]. Все числа вещественные, значения *x*, *a* и *b* вводятся с клавиатуры. Разработайте два варианта программы: с использованием вложенных условных опера-торов и со сложным условием.

2. Напишите программу, которая получает четырёхзначное число и определяет, верно ли, что оно является палиндро-мом (программа должна вывести ответ «да» или «нет»). На-пример, число 2332 – палиндром, а 2342 – нет.