Each person p must be assigned with 223 working days in the year, outside the already assigned 30 vacation days. At each working day, each person must be assigned with one of two possible shifts s, with s = 1 (Morning) or 2 (Afternoon), and one of two possible teams.

The set of possible teams to be assigned to person p is T_p (and $|T_p|$ indicates the number of possible teams of person p). Moreover, set P_1 is set of persons such that $|T_p| = 1$, while set P_2 is set of persons such that $|T_p| = 2$.

A timetable is defined by *Table*. In the algorithm, *Table* + (p,d,s,t) means to include in the timetable the assignment of the working day d to person p in shift s and team t.

A timetable must be compliant with the following three rules:

1. Each person cannot be assigned with more than 5 working days in sequence.
2. Each person cannot be assigned with more than 22 days on Sundays and Bank Holidays.
3. Each time a person is assigned with a working day d in the Afternoon shift, it cannot be assigned with the next working day d+1 in the Morning shift.

In the algorithm, the function f_1(*Table*,p,d,s) checks if assigning the working day d and shift s to person p in the current timetable *Table* violates these three rules. It returns TRUE if none of the rules is violated, or FALSE otherwise.

The input data defines a minimum and an ideal number of persons to be assigned to each shift s and each team t of each day d of the year. In the algorithm, the function f_2(*Table,d,s,t*) returns a non-negative integer that depends on the current number n of persons in *Table* assigned with day d, shift s and team t:

- If n is below the required minimum number of persons, f_2(*Table,d,s,t*) returns 0.
- If n is at least equal to the minimum number of persons and below the ideal number of persons, f_2(*Table,d,s,t*) returns 1.
- If n is at least equal to the ideal number of persons, f_2(*Table,d,s,t*) returns 2 plus the number of persons above the ideal number.

In the algorithm, *numIter* is a parameter that implements a trade-off between running time and quality of the final timetable solution. Preliminary tests show that *numIter* = 10 is a good parameter value.

Build a solution:

1. *Table* ← an empty timetable
2. **While** *Table* not complete **do**
3. P ← set of persons in P_1 without a complete timetable
4. **If** P is empty **do**
5. P ← set of persons in P_2 without a complete timetable
6. **EndIf**
7. p ← a person randomly selected from P
8. *f_value* ← +inf
9. *count* ← 0
10. **While** *f_value* > 0 **and** *count* < *numIter* **do**
11. d ← a day randomly selected from the possible working days still not assigned to person p
12. s ← a randomly selected shift
13. **If** $f_1(Table,p,d,s)$ is TRUE **do**
14. *count* ← *count* + 1
15. **For** $t \in T_p$ **do**
16. *f_aux* ← $f_2(Table,d,s,t)$
17. **If** *f_aux* < *f_value* **do**
18. *f_value* ← *f_aux*
19. $(p,d,s,t)_{best} \leftarrow (p,d,s,t)$
20. **EndIf**
21. **EndFor**
22. **EndIf**
23. **EndWhile**
24. *Table* ← *Table* + $(p,d,s,t)_{best}$
25. **EndWhile**