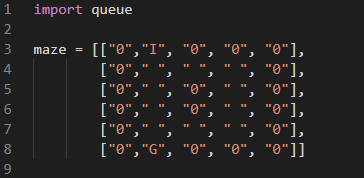
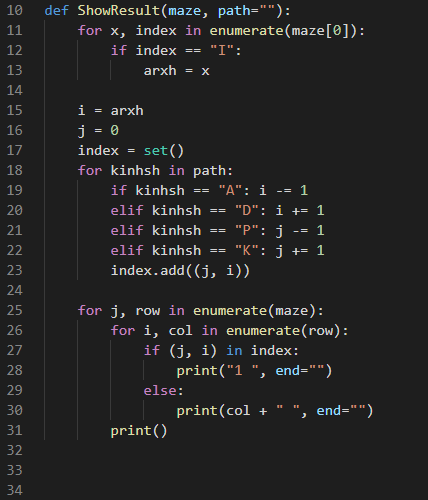
**BREADTH BRANCH AND BOUND**

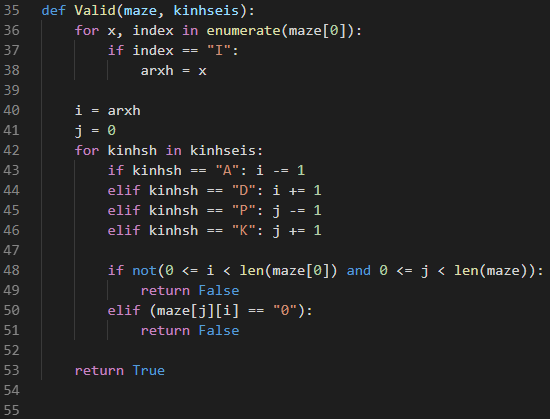
Αρχικά κάνω import τη βιβλιοθήκη "queue" η οποία θα χρησιμοποιηθεί για το προγραμμά μας αργότερα. Δημιουργώ τη λίστα “maze” στη οποία θέτω ως "I" το σημείο εκκίνησης, ως "G" το σημείο τερματισμού, " " στη διαδρομή από την οποία μπορούμε να περάσουμε και "0" εκεί που δεν μπορούμε.



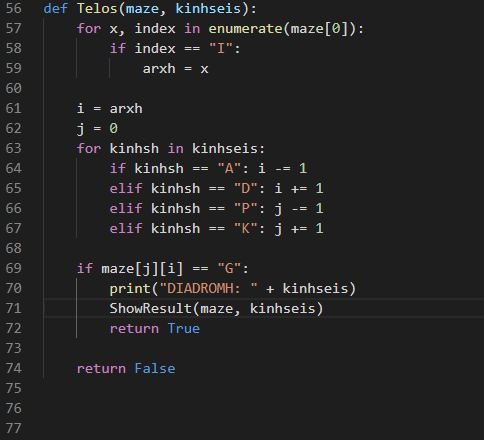
Παρακάτω με το function ShowResult θα δείχνω τον ήδη λυμένο λαβύρινθο τοποθετώντας “1” στα σημεία της καλύτερης διαδρομής. Θέτω τις κινήσεις “A”, ”D”, “P”, “K” (Aristera, Deksia, Panw, Katw) τις οποίες χρησιμοποιώ με τις ίδιες ονομασίες και στο υπόλοιπο μέρος του προγράμματος.



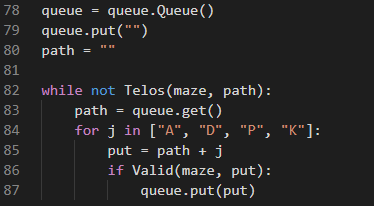
Με το function Valid ελέγχω εάν είναι έγκυρη η επόμενη κίνηση ελέγχοντας αν το i και το j είναι μέσα στα πλαίσια της λίστας κάνοντας return False αν δεν είναι ή εάν η τιμή σε αυτό το σημείο είναι "0".



Με το function Telos ελέγχω αν με τη διαδρομή μου έχω φτάσει στο τελικό σημείο G και σε αυτή την περίπτωση επικαλούμαι το ShowResult και δείχνω το αποτέλεσμα.



Σε αυτό το σημείο επικαλούμαι το queue (σειρά) και ξεκινάω την επανάληψη η οποία επαναλαμβάνεται εώς ότου λυθεί το πρόβλημα. Προφανώς χρησιμοποιώ και το function Valid για να ελέγξω αν οι τιμές είναι έγκυρες.



Το αποτέλεσμα του αλγορίθμου στον τρίτο λαβύρινθο της εργασίας:

