Das Programm wird in Python implementiert. Die Verteilung der Wünsche funktioniert wie folgt:

- Die Erstwünsche, die in ersten Position nur einmal auftauchen, werden zuerst verteilt.
- Es ist dabei wichtig, dass die vergebenen Geschenke nicht mehr in Wunschliste auftreten. Sie werden dann auf 0 gesetzt.
- Außerdem haben die Wünsche, die in ersten Position stehen, einen höheren Wert als in niedrigeren Positionen, das heißt, wir geben sie lieber den Kindern, die sie sehr mögen, als den Kindern, die sie “weniger” mögen. Aus dem Grund werden die Wünschen in zweiten und dritten Position auf 0 gesetzt, wenn sie Erstwünsche sind. Das gleiche Prinzip gilt auch für Zweitwünsche in dritten Position.
- Nunmal betrachten wir die Wünsche in ersten Position. Sie tauchen mehrmals auf, aber zwischen 2 Schülern, dem einen nur diesen Wünsch übrig hat, weil seine anderen Wünsche schon auf 0 gesetzt wurden, und dem anderen, der noch andere Möglichkeit hat, wäre es praktischer, dem ersten Kind diesen Wunsch zu geben, z.B. zwischen (30, 0, 0) und (30, 0, 3), wir sehen, dass wenn 30 dem ersten Kind geschenkt wird, hat der zweite immerhin 3 als Wunsch, nur nicht so krass gewollt ist. Aber wenn der zweite 30 bekommt, muss der andere irgendein zufälliges Geschenk bekommen. Die Vorgehensweise wird dann implementiert, indem wir schauen, welches Kind nur den einen Wunsch hat und gibt dem diesen. Es kann passieren, dass 2 Kinder das Gleiche hat, z.B (30, 0, 0) und (30, 0, 0). Hier muss leider der zweite ein anderes bekommen. Das gleiche Prinzip gilt auch für Zweitwünsche.
- Letztendlich geben wir die Wünsche in dritten Position aus.
- Die unzufriedenen übrig gebliebenen Kinder bekommen dann die übrig gebliebenen Geschenke.

1. **Einlesen des Daten**

list1 beinhaltet die abgelesenen Daten im String-Format und mit Zeilensprungzeichen (\n). In kids_wishes werden die Elemente gefiltert.

2. **Deklaration wichtiger Variablen**

Z.9+10: Jede 3 Wünsche wird numeriert von 1. Danach werden die Listen, die von enumerate() gebildet wurden, ausgepackt, damit haben z.B wir noch übrig:

```
[(1, 2, 10, 6), (2, 2, 7, 3), (3, 4, 7, 1), (4, 3, 4, 9), (5, 3, 7, 9), (6, 4, 3, 2),(7, 7, 6, 2), (8, 10, 2, 4), (9, 9, 8, 1), (10, 4, 9, 6)]
```

Da präsentiert die erste Zahl jeder Liste den Schüler.

Z.12: gifts gibt die Wichtelgeschenke wieder, welche eigentlich die Nummer der Schüler sind.

Z.18: one_wanted sind die Erstwünsche, die nur einmal gewollt sind.
Z.19: more_wanted sind die Erstwünsche, die mehrmals gewollt sind.

3. **Methoden**

Z. 15: nth_wishes() gibt die Wünsche in einer beliebigen Position wieder.

Z.22: update() setzt nun die Werte, die ausgegeben wurden, auf 0.

Z.36: give_gifts() wie der Name schon sagt, verteilt die Wünsche an die Kinder. Es fällt auf, dass diese Methode eine andere Funktion als Übergabe bekommt. Sie ist und zwar die Voraussetzung, welche Geschenke dann ausgegeben werden sollen, basierend auf der oben geklärten Vorgehensweise.

Z.52: wichteln() ist die Hauptmethode.

- x ist zuerst die Liste, nachdem die einmal gewünschten Erstwünsche verteilt worden sind.
- y ist die Liste, indem die Wünsche, die niedrigeren Rang haben, auf 0 gesetzt werden.
- z ist die Liste, nachdem die Kinder, die nur einen Wunsch übrig haben, ihren Wünsche bekommen. (bzw. (30, 0, 0) → er bekommt 30).
- w stellt theoretisch das gleiche wie z dar, aber hier bezieht es sich auf die Wünsche auf 2. Position. (z.B (0, 25, 0) → er bekommt 25) Und bei v werden die Drittwünsche verteilt.
- Nun haben wir die übrigen Kinder und die übrigen Geschenke left_kids und left_gifts. Die Elemente von diesen beiden Listen geben wir einfach in Paare wieder.

Anmerkung:

Das Konzept der Verteilung ist einfach zu verstehen, allerdings ist die Implementierung in in den Code ziemlich kompliziert.