

# Zadanie Bajt Trade

Celem tego zadania będzie stworzenie symulacji rynku, w którym biorą udział **Agenci**, których celem jest zdobycie jak największej liczby diamentów. Wyróżniamy dwa podstawowe typy Agentów: **Robotników** i **Spekulantów**. W razie jakichkolwiek wątpliwości proszę spojrzeć na sekcję założenia.

## Przebieg symulacji

Symulacja będzie odbywać się w turach (nazywanych też dniami). Symulację zaczynamy od tury numer 1. Każda tura składa się z następujących etapów wykonywanych w następującej kolejności. Każdy z etapów jest szczegółowo opisany w odpowiedniej sekcji:

- 1) Robotnicy decydują, czy poświęcają dany dzień na uczenie się, czy na pracę.
  - a) Robotnik się uczy. Po nauce i kończy dany dzień (nie odnoszą się do niego żadne punkty poniżej).
  - b) Robotnik pracuje.
    - i) Robotnik na podstawie swoich strategii produkuje przedmioty.
    - ii) Robotnik wystawia na giełdzie oferty sprzedaży produktów, które wyprodukował. Oferty sprzedaży Robotników nie mają ceny, tylko liczbę i typ przedmiotów, które chcą sprzedać.
    - iii) Robotnik wystawia oferty kupna. Podobnie, oferty kupna nie mają ceny, tylko typ oraz liczbę przedmiotów.
- 2) Na giełdę wchodzi Spekulant z swoimi ofertami kupna i sprzedaży. Ich oferty mają już (poza liczbą i typem produktów) cenę.
- 3) Giełda wykonuje proces dopasowywania ofert kupna z ofertami sprzedaży. Dopasowane zostaną tylko oferty Robotników z ofertami Spekulantów tym samym ustalając cenę transakcji. Zatem Robotnik nie sprzedaje/kupi nic innemu Robotnikowi. Podobnie Spekulant.
- 4) Po realizacji dopasowanych transakcji Giełda skupuje niesprzedane produkty od Robotników. Niezrealizowane oferty Spekulantów (kupna i sprzedaży) i Robotników (kupna) są czyszczone z pamięci Giełdy. Zatem każdą turę Giełda zaczyna bez ofert.
- 5) Na koniec dnia Robotnicy zużywają swoje przedmioty: jedzenie, ubrania i narzędzia oraz **wykorzystane** programy komputerowe.

Warto zwrócić uwagę, że uczący się Agent nie zużywa żadnych przedmiotów pod koniec tury, **również jedzenia**.

## Robotnicy

Każdy Robotnik zaczyna symulację z ustaloną liczbą produktów (wczytaną z wejścia). Dodatkowo każdy Robotnik ma przypisany na stałe wektor bazowy produktywności, który dla każdego produktu określa, ile jednostek tego produktu Robotnik jest w stanie wyprodukować w jednej turze. Liczby w tym wektorze **zawsze będą wielokrotnościami 100**. Podczas symulacji ta wartość będzie modyfikowana przez różne premie (wyrażone w całkowitej

liczbie procentów, np.: +10%, +20%, -1%, -150%), więc Robotnicy zawsze będą produkowali całkowitą liczbę produktów na turę. Jeżeli po dodaniu premii (mogą być ujemne) wyjdzie, że produkujemy ujemną liczbę przedmiotów to nic nie produkujemy. Przykładowo: bazowa produkcja narzędzi to 200 i mamy następujące premie -10%, 20%, 40%. W danej turze wyprodukujemy  $200 + 200 * (20 + 40 - 10)\% = 200 + 100 = 300$ .

W każdej turze Robotnik może albo się uczyć albo pracować. Uczenie polega na awansie w aktualnej ścieżce kariery albo zmianie ścieżki kariery. Praca polega na produkcji, sprzedaży oraz kupnie produktów. Jeżeli Robotnik w danej turze pracuje to dodatkowo zużywa (traci) przedmioty w następujący sposób:

- Zużywa 100 jednostek jedzenia. Jeżeli ma mniej to zużywa tyle ile ma, ale liczy się tak jakby tego dnia nic nie jadł.
- Zużywa wszystkie swoje narzędzia.
- Dodaje jeden dzień noszenia do dowolnych 100 ubrań. Pozostałe ubrania (jeżeli ma) pozostają bez zmian. Dane ubranie będzie całkowicie zużyte, jeżeli było noszone tyle dni, ile wynosi jego wytrzymałość. Jeżeli Robotnik ma mniej niż 100 ubrań to dodaje dzień noszenia do wszystkich jakie ma.
- Zużywa te programy komputerowe, których użył do produkcji w danym dniu. Pozostałe zostają bez zmian.

## Kiedy robotnik się uczy, a kiedy pracuje?

Każdy Robotnik ma jedną z następujących strategii, które determinują co robi danego dnia:

- **Pracuß** nigdy się nie uczy, zawsze pracuje.
- **Oszczędný** uczy się wtedy i tylko wtedy, gdy ma powyżej limit\_diaamentów  
diaamentów, gdzie limit\_diaamentów to parametr strategii.
- **Student** uczy się zawsze wtedy i tylko wtedy, gdy stać go na kupno  $100 * \text{zapas}$  jednostek jedzenia licząc po cenie równej średniej arytmetycznej średnich cen przez ostatnie okresy dni, gdzie zapas i okres to parametry strategii.
- **Okresowy** uczy się co okresowosc\_nauki dni, gdzie okresowosc\_nauki to parametr strategii. W pozostałe dni pracuje. **Przykładowo:** dla okresowosc\_nauki = 10, będziemy się uczyć w turze 10, 20, 30, itd.
- **Rozkładowy** z prawdopodobieństwem  $1 - 1/(\text{dzień symulacji} + 3)$  pracuje, z prawdopodobieństwem  $1/(\text{dzień symulacji} + 3)$  uczy się.

## Jak uczy się Robotnik?

Robotnik uczy się na jeden z 2 sposobów: albo rozwija się w aktualnej ścieżce albo zmienia ścieżkę. Po zmianie ścieżki kariery Robotnik nie zapomina poziomu poprzednich ścieżek kariery. Co prawda może korzystać z premii do produkcji tylko z aktualnej ścieżki, ale jak zmieni ścieżkę na taką, którą już miał wcześniej, to nie zaczyna od poziomu 1, tylko od tego, który miał. Przykładowo: Rolnik poziomu 3, stał się Górnikiem. Nigdy wcześniej nie był Górnikiem, więc zostaje Górnikiem poziomu 1. Następnie po pewnym czasie zmienił ścieżkę

z powrotem na Rolnika. Teraz jest Rolnikiem poziomu 3 (zamiast 1), ponieważ taki poziom osiągnął zanim przestał być Rolnikiem.

Robotnicy mają jedną z dwóch strategii zmiany ścieżki kariery:

- **Konserwatysta** nigdy nie zmienia ścieżki kariery.
- **Rewolucjonista** raz na 7 dni oblicza  $n$  jako  $\max(1, \text{swoje id modulo } 17)$ . Następnie wybiera ścieżkę kariery dającą premię odpowiadającą produktowi, który pojawił się najczęściej (w sensie sumarycznej liczby jednostek w ofertach sprzedaży Spekulantów i Robotników) przez ostatnie  $n$  dni. W pozostałe dni nie zmienia ścieżki kariery. Jeżeli w powyższy sposób wybrał ścieżkę kariery, którą aktualnie posiada to zamiast tego się w niej rozwija. Przykładowo: w dniach 1-6 nie zmienia ścieżki, a w 7 turze wybiera potencjalnie nową ścieżkę kariery. Potem w turach 8-13 nie zmienia ścieżki, i w 14 wybiera potencjalnie nową, itd.

## Co produkuje robotnik?

Co produkuje robotnik jest determinowane przez jego strategię produkcji. Mamy kilka takich strategii:

- **Krótkowzroczny** zawsze produkuje produkt, który poprzedniego dnia miał najwyższą średnią cenę (po definicję średniej ceny patrz pod koniec treści).
- **Chciwy** zawsze produkuje produkt, który da jemu w tej turze najwyższy zysk. Zysk liczymy mnożąc liczbę produktów, którą byśmy wyprodukowali przez średnią cenę produktu z poprzedniego dnia.
- **Średniak** zawsze produkuje produkt, którego maksymalna średnia cena w przeciągu ostatnich `historia_sredniej_produkcji` dni była najwyższa, gdzie `historia_sredniej_produkcji` to parametr danego robotnika.
- **Perspektywiczny** zawsze produkuje produkt, którego wzrost cen był najwyższy w przeciągu ostatnich `historia_perspektywy` dni, gdzie `historia_perspektywy` to parametr danego robotnika. Przez wzrost cen rozumiemy różnicę pomiędzy średnią ceną danego produktu teraz i `historia_perspektywy` dni temu.
- **Losowy** zawsze produkuje losowy produkt.

Liczba wyprodukowanych przedmiotów jest określona przez bazowy wektor produktywności Robotnika i przez premię do produktywności. Przykładowo: produkujemy Diamenty. Bazowo 100 dziennie, ale mamy premię 50%, więc wyprodukujemy  $100 + 50\% \cdot 100 = 150$  dziennie.

Dany Robotnik zawsze **produkuje jeden typ** produktu dziennie.

## Co sprzedaje robotnik?

Robotnik sprzedaje wszystko co **wyprodukował** danego dnia, oprócz diamentów. Jeżeli coś kupił wcześniej (lub miał od początku) i mu zostało, to tego **nie sprzedaje**.

Proszę zwrócić uwagę, że oznacza to, że Robotnik nie może jeść jedzenia, nosić ubrać, używać narzędzi oraz programów komputerowych, które sam wyprodukował, ponieważ giełda pod koniec dnia skupuje wszystkie oferty sprzedaży od Robotników.

## Co kupuje robotnik?

Robotnik może mieć jedną z czterech strategii kupowania i stosowania programów komputerowych:

- **Technofob** kupuje tylko 100 jednostek jedzenia dziennie.
- **Czyścioszek** kupuje 100 jednostek jedzenia dziennie oraz dba aby w następnej turze mieć przynajmniej 100 ubrań (biorąc pod uwagę zużywanie ubrań pod koniec tej tury).
- **Zmechanizowany** kupuje 100 jednostek jedzenia, liczba\_narzedzi (gdzie liczba\_narzedzi to parametr strategii) narzędzi dziennie oraz dba o ubrania, tak jak Czyścioszek.
- **Gadżeciarz** kupuje 100 jednostek jedzenia, liczba\_narzedzi (gdzie liczba\_narzedzi to parametr strategii) narzędzi dziennie oraz dba o ubrania, tak jak Czyścioszek. Dodatkowo zawsze kupuje tyle programów, aby mieć ich tyle, ile wyprodukował produktów w aktualnej rundzie. Programy używa jak tylko może w kolejności od tych z największym poziomem. **Przykładowo:** Mamy 5 programów poziomu 2 i 3 programy poziomu 1. Produkujemy narzędzia w liczbie 6. Wyprodukujemy 6 narzędzi: pięć z poziomem jakości 2 i jeden z poziomem produkcji 1. Po zakończeniu dnia pozostaną nam 2 programy poziomu 1, bo ich nie zużyliśmy.

## Ścieżki kariery

Dostępne są następujące ścieżki kariery:

- **Rolnik**
- **Górnik**
- **Rzemieślnik**
- **Inżynier**
- **Programista**

Każda ścieżka może mieć dowolny dodatni poziom, który daje premię do produkcji.

Poziom	premia
1	+50% wartości bazowej
2	+150% wartości bazowej
3	+300% wartości bazowej
każdy kolejny poziom	+300% + kumulujące się +25% za każdy poziom powyżej 3. Procenty odnoszą się do wartości bazowej.

Premia do produkcji jest naliczana tylko do produktu związanego z daną ścieżką rozwoju:

Rolnik	Jedzenie
--------	----------

Rzemieślnik	Ubrania
Inżynier	Narzędzia
Górnik	Diamenty
Programista	Programy komputerowe

## Produkty

W symulacji dostępne są następujące produkty:

- **Diamenty**
- **Jedzenie**
- **Ubrania**
- **Narzędzia**
- **Programy komputerowe**

Każdy produkt ma swoje unikatowe zastosowania:

Jedzenie	Jest niezbędne każdemu <b>pracującemu</b> robotnikowi i każdy <b>pracujący</b> robotnik zużywa 100 jednostek jedzenia dziennie. Jeżeli Robotnik nie jadł poprzedniego dnia dostaje premię do wydajność -100%, jeżeli nie jadł dwa dni z rzędu dostaje zamiast tego premię -300%. Po trzecim dniu Robotnik umiera. Robotnik martwy nie dokonuje już żadnych działań, a jego wynik gry wynosi zero diamentów. <b>Jeżeli danego dnia Robotnik się uczył to zakładamy, że zjadł na stołówce za darmo i zerujemy licznik, ile dni z rzędu nie jadł.</b>
Ubrania (mają poziomy jakości)	Posiadanie mniej niż 100 ubrań na początku tury powoduje że Robotnik dostaje premię do produktywności równą -kara_za_brak_ubrań%, gdzie kara_za_brak_ubrań jest parametrem symulacji. Ubrania zużywają się po $y^2$ turach używania, gdzie y to poziom jakości ubrania.
Narzędzia (mają poziomy jakości)	Każde narzędzie daje premię do produktywności +y% (jak mamy wiele narzędzi to premie się sumują), gdzie y to poziom jakości narzędzia. Narzędzia po danej turze (w której produkujemy) się całkowicie zużywają.
Diamenty	Są ostateczną walutą w grze.

Program komputerowy (mają poziomy zaawansowania)	Pozwala ustawić przy produkcji narzędzi i ubrań ich poziom jakości na poziom zaawansowania programu. Jeżeli robotnik jest Programistą to poziom zaawansowania programu to jego poziom ścieżki kariery, a jeżeli nie to 1. Jeden program działa na jedno narzędzie lub ubranie. Programy są jednorazowe, więc zużywają się po ich użyciu.
--	--

## Opis giełdy

Na giełdzie pojawiają się oferty kupna i sprzedaży (z ceną i liczbą jednostek od Spekulantów i bez ceny, ale z liczbą jednostek od Robotników). **Na giełdzie nie można sprzedawać/kupować diamentów!** W przypadku programów komputerowych, **narzędzi oraz ubrań** mamy osobne oferty dla różnych poziomów zaawansowania programów. Wyjątek stanowi oferta kupna programu przez Robotnika - on chce tylko kupić ileś programów, **narzędzi, ubrań, z najwyższym poziomem jakości/zaawansowania**, jakie są dostępne.

Na giełdzie wymiana może zachodzić według jednej z trzech strategii:

- **Kapitalistyczna** gdzie najpierw realizowane są transakcje Robotników mających najwięcej diamentów,
- **Socjalistyczna** gdzie kolejność jest odwrotna,
- **Zrównoważona** gdzie w kolejnych turach następują po sobie strategia socjalistyczna i kapitalistyczna.

W przypadku takiej samej liczby diamentów o kolejności decyduje id robotnika. Innymi słowy porównujemy pary po współrzędnych (liczba diamentów, id robotnika) raz w kolejności rosnącej, a raz w malejącej.

Robotnik gdy przyjdzie jego kolej dokonuje transakcji według najkorzystniejszej z ofert dostępnych na rynku. Robotnik realizuje od razu wszystkie swoje zlecenia, **najpierw wszystkie sprzedaży potem wszystkie kupna**. Oferty Spekulantów i Robotników mogą zostać zrealizowane częściowo. Kolejność sprzedaży i kupna produktów dla danego Robotnika jest zawsze taka sama: **Jedzenie, Ubrania, Narzędzia, Programy Komputerowe**. W przypadku programów komputerowych, ubrań i narzędzi oferty **sprzedaży** Spekulantów są sortowane jako pary (poziom zaawansowania/**jakości nierosnąco**, cena **niemalejąco**). Oznacza to, że korzystniejsza oferta to **przykładowo** program wyższego poziomu niezależnie od ceny, a w przypadku programów tego samego poziomu o korzystności oferty decyduje cena.

**Przykładowo:** dany Robotnik chce sprzedać 5 **jednostek jedzenia**, a oferty kupna Spekulantów na giełdzie to (1 **jedzenie** po 3 diamenty), (100 **jedzenia** po 4 diamenty) i (3 **jedzenia** po 5 diamentów). Robotnik zarobi  $5 * 3 + 4 * 2 = 23$  diamenty. Na giełdzie

pozostaną oferty (1 **jedzenia** po 3 diamenty) i (98 **jedzenia** po 4 diamenty). Ten sam Robotnik chce kupić 4 narzędzia (**nie podaje ich poziomu**). Oferty Spekulantów to (5 narzędzi **poziomu 3** po 29 diamentów) **oraz (100 narzędzi poziomu 2 po 1 diament)**. Zakładając, że Robotnik miał wcześniej 7 diamentów to **po sprzedaży** ma równo 30, więc stać go tylko na **2** narzędzia, **jedno poziomu 3 (bo najpierw kupuje najlepsze) i jedno poziomu 2**. Zatem na giełdzie zostaną oferty (4 narzędzia **poziomu 3** po 29 diamentów) **i (99 narzędzi poziomu 2 po 1 diament)**. Teraz przechodzimy do realizacji ofert następnego Robotnika w kolejności.

Jeżeli Robotnikowi nie uda się sprzedać tego, co chciał (ponieważ skończyły się oferty kupna Spekulantów), to resztę produktów na sprzedaż kupuje bank po najniższej cenie kupna dla danego produktu z poprzedniego dnia (tury) lub cenie z tury “zerowej”, jeżeli nie było ofert kupna dla danego produktu poprzedniego dnia.

Niezrealizowane oferty kupna i sprzedaży Spekulantów oraz oferty kupna Robotników znikają, nie zostają na następną turę.

## Spekulanci

Drugim typem Agenta jest **Spekulant**. Zamiast produkcji handluje on na giełdzie. Sam nie produkuje, ~~**ale ma ustalony budżet diamentów na turę, za które może kupować przedmioty. Niewykorzystany budżet się marnuje i odnawia całkowicie na początku następnej tury.**~~ Na początku spekulant nie ma żadnych przedmiotów. Opis giełdy jest w następnej sekcji.

Każdy spekulant, jeżeli wystawia ofertę kupna, to kupuje 100 jednostek danego produktu (**programy, ubrania i narzędzia różnych poziomów traktujemy tutaj jako różne produkty**). Jeżeli wystawia ofertę sprzedaży, to sprzedaje wszystko co ma w jednej ofercie. Spekulanci mają różne strategie handlu (taką samą dla każdego rodzaju produktu i poziomu zaawansowania programu **oraz poziomu jakości ubrań i narzędzi**, od 1 do numeru tury symulacji):

- **Spekulanci średni** wystawiają oferty **kupna i sprzedaży** 10% poniżej i 10% powyżej średniej arytmetycznej ze średnich cen z ostatnich historia\_spekulanta\_sredniego (parametr spekulanta) dni. Jeżeli dany spekulant nie ma danego produktu, to zamiast tego daje **tylko** ofertę kupna 5% poniżej tej wartości. **Spekulant średni** co turę sprzedaje wszystko co ma (oczywiście poza diamentami, patrz opis giełdy).
- **Spekulanci wypukli** kupują wyłącznie, gdy funkcja utworzona ze średnich cen ostatnich 3 dni jest ściśle wypukła, a sprzedaje jak jest ściśle wklęsła. W innym przypadku nie robi nic. Ceny kupna i sprzedaży to odpowiednio 10% poniżej i 10% powyżej średniej ceny z ostatniego dnia.
- **Spekulant regulujący rynek** w pierwszej turze nic nie robi. W następnych bierze średnią cenę z poprzedniego dnia i mnoży ją w turze  $i$  przez  $p_i / \max(p_{i-1}, 1)$ , gdzie  $p_i$  to liczba produktów danego rodzaju wystawiona do sprzedaży przez robotników w turze  $i$ . Następnie wystawia ofertę kupna 10% poniżej tej ceny i sprzedaży 10% powyżej.



# Json

Do obsługi zapisu i odczytu formatu JSON można wykorzystać bibliotekę moshi:

<https://github.com/square/moshi>

lub Gson

## Dodatkowe założenia i definicje

- W przypadku dowolnego patrzenia wstecz x tur: Jeżeli od początku symulacji upłynęło mniej niż x pełnych tur, to po prostu bierzemy ile jest. Na wejściu pojawiają się koszty wszystkich produktów (poza diamentami) w turze “zerowej”. Interpretujemy to tak, że w “zerowej” turze była jedna oferta kupna Spekulanta dla każdego produktu (1 produkt za ... diamentów). Cena dla różnych poziomów zaawansowania programów komputerowych jest jedna dla wszystkich w turze “zerowej”. **Tak samo cena dla różnych poziomów jakości ubrań/narzędzi**. Przykładowo: średnia z 10 tur potrzebna do podjęcia decyzji w 5 turze to średnia z pierwszych 5 tur (4 prawdziwych i sztucznej zerowej), tak samo odnośnie liczenia funkcji wypukłych/wklęsłych. Jeżeli jednak w turze 100 liczymy średnią z 5 ostatnich dni, to bierzemy tylko 5 ostatnich dni, zerowa tura nie ma wtedy znaczenia.
- Zakładamy, że diamenty są idealnie podzielne tak jak bitcoiny.
- Id Robotników są parami różne.
- Przez średnią cenę z danego dnia rozumiemy średnią ważoną cen z wagami będącymi liczbą przedmiotów. Pod uwagę bierzemy **wszystkie** oferty **kupna i sprzedaży** Spekulantów, ale tylko te zrealizowane. Jeżeli nie było żadnych zrealizowanych ofert danego produktu to średnia cena to cena z tury “zerowej” danego produktu. Przykładowo: mamy oferty (5 ubrań po 1 diament) oraz (2 ubrania po 3 diamenty). Średnia cena ubrań tego dnia to  $(5 * 1 + 2 * 3) / 7$ .
- **Przedmioty posiadane przez agentów na początku symulacji (podane na wejściu) są traktowane jako przedmioty poziomu 1 (dla ubrań, narzędzi i programów) oraz jako kupione. Oznacza to, że ich nie sprzedajemy, tylko z nich korzystamy.**

## FAQ

- Czy Robotnik zużywa jedzenie/narzędzia/ubrania/programy komputerowe w turze, kiedy się uczy? Odp.: **Nie**
- Co się dzieje z niewykorzystanymi programami komputerowymi? Przykładowo produkuje 10 narzędzi, ale mam 100 programów komputerowych? Odp.: **Pod koniec tury Robotnik traci 10 programów (bo tyle użył do produkcji narzędzi), ale 90 mu zostaje i może ich użyć w następnych turach.**
- Co dokładnie oznacza zużycie przedmiotu? Odp.: **Oznacza, że taki przedmiot powinien być usunięty z symulacji w momencie jego użycia. Nie ma przedmiotów o statusie “zużyty”, wszystkie takie idą do “kosza”.**



- Co zrobić przy wyborze produktu/ścieżki kariery, w przypadku remisu (według ustalonej miary)? Odp.: **Remisy rozstrzygamy biorąc pod uwagę kolejność ścieżek kariery/produktów w tabelce w treści. Przykładowo, jeżeli chcemy produkować i według naszej strategii Narzędzia i Programy komputerowe są równie dobre, to produkujemy Programy komputerowe, ponieważ są dalej w tabelce. Analogicznie, jeżeli chcemy zmienić ścieżkę kariery i Rolnik oraz Rzemieślnik są równie dobre, to wybieramy dalszą w tabelce, czyli Rzemieślnika.**

## Format rozwiązania

Jako rozwiązanie należy wysłać jeden plik (archiwum zip). W środku powinny znajdować się wszystkie pliki i katalogi niezbędne do skompilowania rozwiązania na systemie **students**. W szczególności powinna być zachowana struktura folderów odpowiadająca strukturze pakietów w rozwiązaniu. Oprócz źródeł projekt może zawierać pliki konfiguracyjne, używane do ściągnięcia i zainstalowania dodatkowych bibliotek (chodzi głównie o te do czytania/pisania Jsonów). Polecamy do tego narzędzie Maven. W przypadku użycia innego narzędzia, proszę zamieścić w rozwiązaniu plik README.md krótko opisujący, jak skompilować i uruchomić Państwa rozwiązanie.

Prosimy również zamieścić w paczce 2 skrypty bashowe o nazwie `kompiluj.sh` oraz `uruchom.sh` (proszę się upewnić, że mają dobrze ustawioną flagę `executable`, jak nie to można to naprawić pisząc `chmod +x kompiluj.sh` oraz `chmod +x uruchom.sh`). Pierwszy skrypt powinien kompilować Państwa rozwiązanie, a drugi je uruchamiać. Skrypt `uruchom.sh` powinien przyjmować 2 argumenty: lokalizację pliku wejściowego i wyjściowego. Plik wyjściowy powinien zostać stworzony, jeżeli nie istnieje (foldery na ścieżce do niego będą już stworzone, czyli przykładowo przy uruchomieniu `uruchom.sh inputs/in.json tests/students/out.json`, można założyć, że foldery `tests/students` już istnieją). Nazwy plików mogą zostać przekazane przez skrypt do Państwa programu w dowolny sposób (jako argumenty funkcji `main`, jako `stdin`, czy inaczej). Prosimy jednak podawać programowi nazwy plików, a nie ich zawartość (np.: przez przekierowanie `stdin/stdout` do podanych plików).

## Wejście i Wyjście

Patrz przykładowe pliki `wejście.json` i `wyjście.json`.

Proszę zwrócić uwagę, że na wejściu wypisujemy liczby narzędzi/ubrań/programów według poziomów w tablicy. Nie wypisujemy liczby posiadanych rzeczy danego poziomu, jeżeli ta liczba wynosiłaby 0 (chyba, że potrzebujemy wypisać to 0, żeby móc wypisać liczbę rzeczy wyższych poziomów - patrz przykład).