

Etap	1	2	3	4	Suma
Punkty	4	3	5	4	16
Wynik					

## L2: Przedszkole

Zadanie polega na napisaniu programu w standardzie POSIX, który symuluje rozprzestrzenianie się wirusa grypy w przedszkolu. Główny proces powinien odpowiadać za obsługę przedszkola, a  $n$  procesów potomnych za symulowanie dzieci przebywających w przedszkolu. Symulacja trwa  $t$  sekund.

Gdy dziecko jest chore, zaczyna kasłać co pewien czas (losowo, co 50-200 ms) i kaszle na wszystkie inne dzieci w przedszkolu – wysyłając sygnał `SIGUSR1`. Gdy zdrowe dziecko otrzyma wirusa (sygnał), istnieje prawdopodobieństwo  $p$ , że samo zachoruje i zacznie kaszleć. Po chore dziecko przyjeżdżają rodzice po  $k$  sekundach i je zabierają.

Program jako argumenty powinien przyjąć wartości  $t$ ,  $k$ ,  $n$  oraz  $p$ , gdzie:

- $t$  - to czas symulacji w sekundach (1 - 100),
- $k$  - to czas po jakim przyjeżdżają rodzice po chore dziecko (1 - 100),
- $n$  - to liczba dzieci w przedszkolu (1 - 30),
- $p$  - to wartość prawdopodobieństwa wyrażona w procentach, że dziecko po kontakcie z wirusem zachoruje (1 - 100).

Należy założyć, że zawsze jedno dziecko jest chore na starcie, dla uproszczenia można przyjąć, że jest to pierwsze z listy.

Główny proces po czasie  $t$  sekund wysyła `SIGTERM` do wszystkich dzieci, który oznacza zakończenie symulacji. Dziecko po otrzymaniu sygnału, kończy swoje działanie i zwraca jako exit status liczbę kaszlnięć (aby przekazać ją do głównego procesu). W celu odliczania czasu użyj funkcji `alarm`.

W momencie odebrania dziecka przez rodziców, dziecko wyświetla o tym fakcie informację oraz kończy swoje działanie wracając odpowiedni exit code. Procesy dzieci, które zakończą się przed wysłaniem sygnału kończącego symulację należy uznać za odebrane.

Gdy wszystkie procesy dzieci zakończą swoje działanie po czasie  $t$ , proces główny wyświetla listę wszystkich dzieci wraz z ich stanem, w poniższym formacie, gdzie Child ID to PID:

```
No. | Child ID | Status
1 | 17487 | Coughed 3 times and parents picked them up!
2 | 17488 | Coughed 1 times and is still in the kindergarten!
3 | 17489 | Coughed 2 times and is still in the kindergarten!
2 out of 3 children stayed at in the kindergarten!
```

Uwaga 1: Należy zadbać, aby funkcje obsługujące sygnały były jak najlżejsze, zawierały jedynie proste operacje. W szczególności niedopuszczalne jest używanie funkcji z grupy `signal-unsafe` (patrz `man 7 signal-safety`)

Uwaga 2: Aby rozsyłać sygnały do wszystkich dzieci użyj ID grupy.

Uwaga 3: Aby uzyskać PID procesu z którego przyszedł sygnał użyj struktury `siginfo_t`.

Wykorzystaj poniższe komunikaty w swojej aplikacji:

TODO:

Przykład działania programu:

```
$ ./sop-kg 10 3 3 20
TODO: add full output
```

Etapy:

1. 4 p. Tworzone są poprawnie procesy dzieci, wypisują swój PID i zasypiają na losowy czas (300-1000 ms), główny proces czeka na ich zakończenie.

Imię i nazwisko:

Procesy, sygnały i deskryptory

2. 3 p. Proces główny wysyła po określonym czasie (alarm w procesie głównym) sygnał SIGTERM do wszystkich dzieci, a dzieci czekają na jego otrzymanie.
3. 5 p. Działa mechanizm kaszlenia - roznoszenia wirusa.
4. 4 p. Dzieci po określonym czasie zostają odebrane przez rodziców (alarm w procesach dzieci).