

# Kolokwium z Programowania Współbieżnego 2016-11-30

Każde zadanie prosimy oddać na osobnej kartce. Prosimy pisać jednostronnie.

## Zadanie 1 (Monitory, MM, 10p.)

W pewnej miejscowości znajduje się boisko do koszykówki, z którego może korzystać  $M$  drużyn koszykarzy oraz gracze niezrzeszeni. Na boisku mogą odbywać się dwa rodzaje rozgrywek:

- mecz – grają dwie drużyny pięcioosobowe,
- gry *jeden na jednego* – na boisku są dwa kosze, czyli mogą grać co najwyżej dwie pary.

Gracz należący do pewnej drużyny (w pętli) zajmuje się własnymi sprawami, a następnie idzie na boisko, gdzie czeka na pozostałych członków swojej drużyny. Po skompletowaniu dwóch drużyn, jeżeli boisko jest wolne, rozpoczyna się mecz. Gracz niezrzeszony, natomiast, (w pętli) po wykonaniu własnych spraw udaje się na boisko, gdzie rozpoczyna grę z dowolnym innym graczem niezrzeszonym, który jeszcze nie rozpoczął gry, o ile nie toczy się w tym czasie mecz (czyli na boisku nie ma żadnego gracza z żadnej drużyny) i jest wolny kosz. Gracze kończą grę pojedynczo. Możemy założyć, że jeżeli na boisku zostanie dwóch graczy niezrzeszonych, to grają ze sobą przy jednym koszu.

Algorytmy graczy są następujące:

```
process GraczBezDrużyny() {
    while (true) {
        pracuje();
        BOISKO.chcęGrać();
        gram();
        BOISKO.skończyłem();
    }
}

process GraczDrużyny(int nrDrużyny) {
    while (true) {
        pracuję();
        BOISKO.poczatekMeczu(nrDrużyny);
        mecz();
        BOISKO.koniecMeczu(nrDrużyny);
    }
}
```

Napisz monitor BOISKO synchronizujący działanie graczy, pamiętając o zachowaniu żywotności.

## Zadanie 2 (Semaforey, KI, 10p.)

Na stoku góry znajduje się  $T$  równoległych torów saneczkowych, numerowanych od 1 do  $T$ , z których korzysta pewna liczba nierozróżnialnych saneczkarzy [process Saneczkarz()]. Każdy saneczkarz w pętli nieskończonej wspina się na górę [void WSPINAM\_SIE()], zajmuje jeden z torów, odpoczywa i podziwia widoki [void ODPOCZYWAM(int tor)], a następnie czeka, aż będzie mógł zjechać i wreszcie zjeżdża [void ZJEZDZAM(int tor)]. Z powodu zmęczenia wspinaczką, saneczkarz zawsze zajmuje pierwszy tor (tj. o najmniejszym numerze), na którym nie odpoczywa ani nie czeka na zjazd inny saneczkarz (w każdym momencie na torze może odpoczywać lub czekać na zjazd co najwyżej jeden saneczkarz). Ponadto, ze względów bezpieczeństwa, saneczkarz nie może zjeżdżać jeśli jego torem lub którymś z torów sąsiednich (o ile istnieje) zjeżdża inny saneczkarz. Wchodzenie na górę, odpoczywanie oraz sam zjazd mogą trwać dowolnie długo. W ekstremalnym przypadku,  $T - 1$  saneczkarzy może zasłabnąć i w efekcie nigdy nie zakończyć wspinaczki lub odpoczywania, w drugim przypadku blokując na zawsze zajęte tory.

Zapisz treść procesu Saneczkarz wykorzystując do synchronizacji semaforey. Twoje rozwiązanie powinno starać się zmaksymalizować liczbę równoległych zjazdów przy zachowaniu żywotności procesów Saneczkarz. Nie może ono korzystać z dodatkowych procesów i musi być zapisane w języku C rozszerzonym o operacje semaforowe – jak na ćwiczeniach. Same pomysły na rozwiązanie – bez zapisu całości w wymaganej notacji – nie będą oceniane.

**Powodzenia!**