Zadanie: MIS
Misie-Patysie

Etap III, dzień I, plik źródłowy `mis.*`, dostępna pamięć 32 MB *31-03-2004*

Bolek i Lolek bawią się w grę o swojsko brzmiącej nazwie misie-patysie. Aby zagrać w misie-patysie wystarczy mieć trochę *misiów* oraz równie nieokreśloną liczbę *patysiów*, które tworzą razem *pulę gry*. Trzeba też wybrać dowolną liczbę naturalną, którą zgodnie z tradycją nazywa się *MP-ograniczeniem*.

Pierwszy ruch należy do Bolka, który może zabrać z puli pewną dodatnią liczbę misiów albo patysiów. Może też jednocześnie wziąć i misie, i patysie, ale tylko jeśli jednych i drugich bierze tyle samo i więcej od zera. Oczywiście nie można zabrać więcej misiów niż jest ich w puli — podobnie z patysiami. Co więcej ani jednych, ani drugich nie można wziąć więcej niż wynosi MP-ograniczenie.

Kolejne ruchy wykonywane są na przemian według tych samych reguł. Wygrywa ten z graczy, po którego ruchu pula pozostanie pusta. Twoim celem jest pomóc Bolkowi w odniesieniu zwycięstwa nad Lolkiem.

Zadanie

Zadanie polega na napisaniu modułu, który po skompilowaniu z odpowiednim programem grającym będzie grał jako Bolek. Na potrzeby tego zadania otrzymasz **uproszczony** program grający, który pozwoli Ci przetestować rozwiązanie. Twój moduł powinien zawierać następujące dwie procedury (funkcje):

- `procedure poczatek (m, p, mpo : LongInt)` lub
  `void poczatek (int m, int p, int mpo)`
  Ta procedura (funkcja) będzie wywołana tylko raz, na początku rozgrywki. Możesz jej użyć by zainicjalizować zmienne lub struktury danych potrzebne podczas rozgrywki. Parametry `m`, `p` oraz `mpo` zawierają odpowiednio liczbę misiów i patysiów w puli gry oraz MP-ograniczenie. Liczby Misiów i Patysiów w puli będą zawsze dodatnie i niewiększe niż 10^7. MP-ograniczenie będzie dodatnie i niewiększe niż 10^6.

- `procedure ruch_bolka (m, p : LongInt; var bm, bp : LongInt)` lub
  `void ruch_bolka (int m, int p, int *bm, int *bp)`
  Ta procedura (funkcja) powinna wyznaczyć kolejny ruch Bolka. Przekazane jej parametry `m` i `p` zawierają odpowiednio liczbę misiów i patysiów w puli gry pozostałych po ostatnim ruchu Lolka. Wartości parametrów przekazanych przy pierwszym wywołaniu procedury (funkcji) `ruch_bolka` są takie same jak odpowiednie wartości parametrów przekazanych do procedury (funkcji) `poczatek`. Przed zakończeniem wykonania procedury zmienne `bm`, `bp` (`*bm` i `*bp` w języku C/C++) powinny zawierać, odpowiednio, liczbę misiów i patysiów wziętych przez Bolka z puli.

Twój moduł nie może otwierać żadnych plików ani korzystać ze standardowego wejścia / wyjścia. Jeśli twój program wykona niedozwoloną operację, w tym np. procedura `ruch_bolka` zwróci ruch niezgodny z zasadami gry, jego działanie zostanie przerwane. W tym przypadku dostaniesz 0 punktów za dany test.

Jeśli Twój program przegra rozgrywkę, dostaniesz 0 punktów za test. Jeśli wygra, dostaniesz maksymalną przewidzianą za dany test liczbę punktów. Dla każdego testu, na którym zostanie uruchomiony Twój program, Bolek może wygrać, niezależnie od ruchów Lolka.

Pliki (Pascal)

W katalogu `mis_pas` znajdziesz następujące pliki:

- `mis.pas` – szkielet modułu grającego zawierający puste procedury `poczatek` i `ruch_bolka`. Powinieneś napisać kod tych procedur.
- `graj.pas` – uproszczony program generujący rozgrywkę. Ruchy Lolka wykonywane są zgodnie z bardzo prostą strategią, natomiast ruchy Bolka wyznaczane są przez wywoływanie procedur modułu grającego `mis.pas`. Możesz użyć tego programu do testowania swojego modułu grającego.

Pliki (C / C++)

W katalogu `mis_c / mis_cpp` znajdziesz następujące pliki:

- `mis.h` – plik zawierający nagłówki funkcji `poczatek` i `ruch_bolka`.
- `mis.c / mis.cpp` – szkielet modułu grającego zawierający puste definicje funkcji zadeklarowanych w pliku nagłówkowym `mis.h`. Powinieneś napisać kod tych funkcji.
- `graj.c / graj.cpp` – uproszczony program generujący rozgrywkę. Ruchy Lolka wykonywane są zgodnie z bardzo prostą strategią, natomiast ruchy Bolka wyznaczane są przez wywoływanie procedur modułu grającego `mis.c / mis.cpp`. Możesz użyć tego programu do testowania swojego modułu grającego.

Rozwiązanie

Wynikiem Twojej pracy powinien być tylko jeden plik `mis.c`, `mis.cpp` lub `mis.pas` ale nie kilka równocześnie.

Przykład

Rozgrywka może przebiegać następująco:

Wywołanie	Opis
`poczatek(7, 2, 3);`	jest 7 Misiów, 2 Patysie, a MP-ograniczenie= 3
`ruch_bolka (7, 2, bm, bp);` lub `ruch_bolka (7, 2, &bm, &bp);`	pierwszy ruch Bolek bierze z puli 1 misia i 1 patysia; zostaje 6 misiów i 1 patyś
`ruch_bolka (3, 1, bm, bp);` lub `ruch_bolka (3, 1, &bm, &bp);`	Lolek wziął z puli 3 misie; zostały 3 misie i 1 patyś Bolek bierze z puli 1 misia; zostają 2 misie i 1 patyś
`ruch_bolka (1, 0, bm, bp);` lub `ruch_bolka (1, 0, &bm, &bp);`	Lolek wziął z puli 1 misia i 1 patysia; został 1 miś i 0 patysiów Bolek bierze z puli 1 misia i wygrywa