

W poniższe pole proszę *bardzo starannie* wpisać swoje imię i nazwisko.

Proszę uważnie przeczytać i podpisać poniższą deklarację.

Oświadczam, że przystępuję do egzaminu poprawkowego z przedmiotu „Programowanie”. Mam świadomość, iż podczas egzaminu nie wolno korzystać z żadnych notatek, książek itp., ani pomocy innych osób i nie wolno odpisywać rozwiązań od innych osób, również w sytuacji, gdy będzie to możliwe bez zwrócenia uwagi egzaminatora. Przyjmuję też do wiadomości, że w razie gdybym oszukiwał i zostało to wykryte (zarówno w czasie egzaminu, jaki i sprawdzania pracy), otrzymam ocenę niedostateczną i sprawa zostanie skierowana do Dziekana.

Podpis zdającego

Proszę nie rozpinać i nie odwracać tych kartek przed ogłoszeniem rozpoczęcia egzaminu.

Egzamin trwa trzy godziny zegarowe. Za każde z czterech zadań można otrzymać od -5 do 25 punktów, zatem za cały egzamin można otrzymać do 100 punktów. Za rozpoczęcie zadania otrzymuje się -5 punktów (dlatego podane w treści punkty za każde zadanie sumują się do 30). Za brak rozwiązania zadania otrzymuje się 0 punktów. Do wyników egzaminu poprawkowego nie dolicza się punktów bonusowych z ćwiczeń. Punkty z egzaminu poprawkowego przeliczają się na oceny następująco: mniej niż 33: ndst, od 33 do 45: dst, od 46 do 58: dst+, od 59 do 71: db, od 72 do 84: db+, od 85: bdb.

Proszę pisać rozwiązania zadań bezpośrednio na kartkach z ich treścią. Uwaga: *nie ma* dodatkowego papieru. Imienia, nazwiska i numeru indeksu proszę **nie** wpisywać na kartkach z rozwiązaniami. Na zakończenie egzaminu proszę oddać do sprawdzenia tylko kartki z rozwiązanymi zadaniami. Brudnopis proszę zatrzymać.

Powodzenia!

Przepisz poniższy program, wstawiając pomiędzy jego wiersze asercje, tak, by można było z nich odtworzyć formalny dowód jego poprawności.

```
{N = n ∧ N ≥ 0}
M = 0;
while (N>0) (
    N = N-1;
    M = M+2;
)
{M = 2n}
```

Programowanie
Egzamin poprawkowy
12 września 2003
Zadanie 2 (25 pkt)

Nr zawodnika: 00

Zdefiniuj następujące metody przekazywania parametrów do procedur: (a) przez nazwę, (b) przez wartość, (c) przez referencję (przez zmienną), (d) przez wynik, (e) przez procedurę.

Opisz abstrakcyjny typ danych *semafor* (o wartościach całkowitoliczbowych), tj. podaj znaczenie wszystkich operacji zdefiniowanych dla tego typu. Używając semaforów zaprogramuj N-elementowy bufor liczb całkowitych (N jest stałą), tj. napisz operacje

```
procedure wstaw (x : integer);  
procedure pobierz (var x : integer);
```

pozwalające wykonywanym współbieżnie procesom na składowanie i pobieranie wartości z bufora. Proces, który próbuje czytać z pustego lub pisać do pełnego bufora powinien zostać uśpiony.

Rozważmy następującą sygnaturę wielorodzajową:

Sorts: $int, list$
Functions: $0 : int$
 $S : int \rightarrow int$
 $+ : int \times int \rightarrow int$
 $nil : list$
 $:: : int \times list \rightarrow list$
 $rev : list \rightarrow list$
 $@ : list \times list \rightarrow list$

Uzupełnij (5 pkt.) ją o odpowiedni zbiór równości, tak, by otrzymać specyfikację, w której 0 oznacza liczbę zero, S — następnik, $+$ — dodawanie, nil — listę pustą, $::$ — operację dołączenia głowy do listy, rev — operację odwracania listy, zaś $@$ — operację spinania list). Napisz (5 pkt.) zasadę indukcji dla gatunków int i $list$ i udowodnij (20 pkt.) przy jej pomocy, że

$$rev(l_1 @ l_2) = rev(l_2) @ rev(l_1)$$