

Egzamin, termin I

imię i nazwisko _____

liczba punktów _____ / 10

dokument klasy B1, archiwizować do 2019-01-01

dyn
10:00

W programie zdefiniowano następujące typy:

```

struct Pracownik
{
    string nazwisko;
    Pracownik * pPrev, * pNext;
    Zadanie * pZadania;
};

struct Zadanie
{
    int priorytet;
    string tresc_zadania;
    Zadanie * pLewy, * pPrawy;
};

```

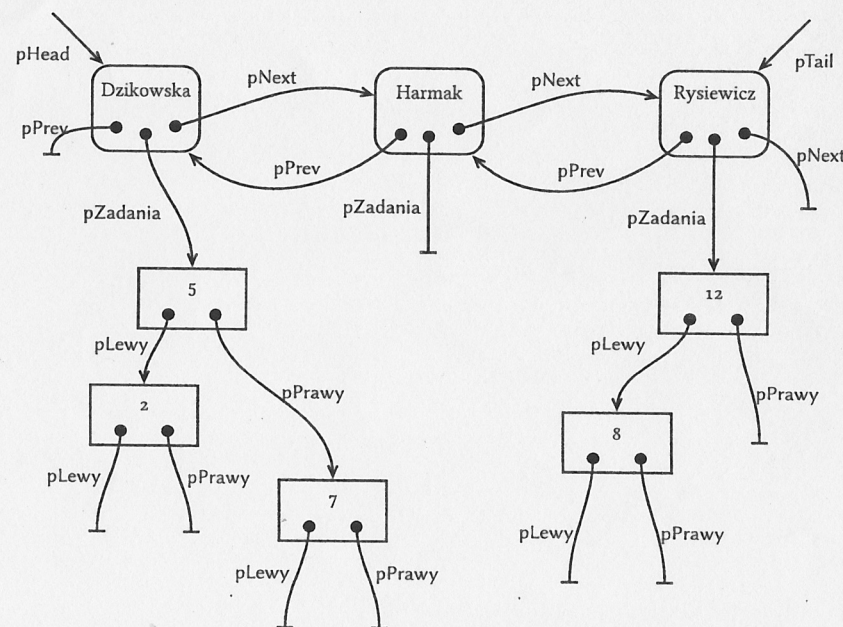
Korzystając z nich można utworzyć strukturę danych, której przykład jest przedstawiony na rys. 1. Pracownicy tworzą listę dwukierunkową uporządkowaną niemalejąco według nazwisk. Zadania przypisane do pracownika tworzą drzewo binarne wg priorytetów zadania. Pracownik może mieć dowolną liczbę zadań (także zero).

Zadanie

- (3 punkty) Proszę zdefiniować funkcję `znajdzLubDodajPracownika`, która odszukuje pracownika o podanym nazwisku. Jeżeli pracownik zostanie znaleziony, to funkcja zwraca adres pracownika. Jeżeli pracownika nie ma w liście, to funkcja dodaje go w odpowiednim miejscu i zwraca adres świeżo dodanego pracownika.
- (1 punkt) Proszę napisać funkcję `dodajZadanie`, która dodaje zadanie do drzewa zadań pracownika. Zadania ułożone są w drzewie według priorytetów zadań.
- (1 punkt) Proszę napisać funkcję `dodajZadaniePracownikowi`, która dodaje zadanie o podanej treści i priorytecie pracownikowi o podanym nazwisku. Jeżeli pracownika nie ma, zostanie on dodany w odpowiednim miejscu. Funkcja korzysta z funkcji zdefiniowanych w punktach 1-2.
- (3 punkty) Proszę napisać funkcję `usunPracownikowBezZadan`, która usuwa wszystkich pracowników, którzy nie mają żadnych zadań. Możliwe, że po usunięciu pracowników bez zadań lista pracowników będzie pusta.
- (2 punkty) Proszę napisać funkcję `odwrocKolejnoscPracownikow`, która odwraca kolejność pracowników, tzn. po jej wykonaniu pracownicy w liście są uporządkowani w odwrotnym porządku niż w liście wejściowej. Po wykonaniu funkcji każdy pracownik zachowuje swoje zadania niezmienione.

Uwaga: Niedopuszczalne jest modyfikowanie definicji struktur `Pracownik` i `Zadanie`.

Proszę pisać czytelnie!



Rysunek 1: Przykładowa struktura danych przechowująca pracowników i ich zadania.