

Laboratorium 6A (7pkt)

Zadanie dotyczy list jednokierunkowych reprezentujących kolejkę studentów w dziekanacie. Kolejka jest reprezentowana przez strukturę `List`, zawierającą wskaźnik do pierwszego elementu listy. Węzeł listy przechowuje dane o studencie (w formie struktury `Student`) oraz jego priorytecie obsługi.

Funkcja `main` jest już gotowa i nie można jej zmieniać, poza komentowaniem i odkomentowywaniem odpowiednich etapów.

W zadaniu nie można tworzyć żadnych list pomocniczych, tablic, itp. – działamy bezpośrednio na podanej liście.

Etap 1 (2 pkt)

Należy zaimplementować funkcje:

- `init_list()` – inicjalizuje pustą listę oraz zwraca wskaźnik na strukturę `list`.
- `add_student(L, student, priority)` – dodaje nowego studenta do kolejki zgodnie z jego priorytetem. Zwraca 1 w przypadku sukcesu, w przeciwnym razie 0. Im wyższy priorytet tym wyżej umieszczamy studenta na liście. W przypadku innego studenta z tym samym priorytetem decyduje kolejność dodania.
- `print_list(L)` – wypisuje całą listę w formacie:
Priority: `priority`, Name `Surname`, index: `index`

Etap 2 (2 pkt)

Należy zaimplementować funkcje:

- `find_student(L, index)` – wyszukuje w kolejce studenta o podanym numerze indeksu i zwraca wskaźnik na jego węzeł. Jeśli student nie zostanie znaleziony, zwraca `NULL`.
- `remove_student(L, index)` – usuwa studenta o podanym numerze indeksu.

Etap 3 (2 pkt)

Należy zaimplementować funkcje:

- `change_priority(L, index, new_priority)` – zmienia priorytet studenta o podanym numerze indeksu i przenosi go na odpowiednie miejsce.
- `reverse_list(L)` - funkcja odwraca kolejność elementów w liście (ostatni student będzie pierwszy, a pierwszy ostatni). Nie można tu tworzyć listy pomocniczej.

Etap 4 (1 pkt)

Należy zaimplementować funkcje:

- `clear_list(L)` – usuwa wszystkie elementy z kolejki.

Oczekiwane wyniki

***** ETAP 1 *****

Priority: 5, Bartosz W., index: 321476
Priority: 4, Piotr S., index: 306179
Priority: 2, Zuzanna O., index: 313095
Priority: 2, Jacek Z., index: 329541

***** ETAP 2 *****

Finding student with a given index

Example 1

Priority: 4, Piotr S., index: 306179

Example 2

Student not found.

Removing student with a given index

Priority: 5, Bartosz W., index: 321476
Priority: 4, Piotr S., index: 306179
Priority: 2, Jacek Z., index: 329541

***** ETAP 3 *****

Changing priority of student

Priority: 5, Bartosz W., index: 321476
Priority: 2, Jacek Z., index: 329541
Priority: 1, Piotr S., index: 306179

Reversing list:

Priority: 1, Piotr S., index: 306179
Priority: 2, Jacek Z., index: 329541
Priority: 5, Bartosz W., index: 321476

***** ETAP 4 *****

Should print empty list