Podstawy programowania

Laboratorium 14

Cel:

- Pogłębianie umiejętności programowania w C

Zadania:

1. Ponowne skopiowanie do osobnego katalogu plików implementacji listy powiązanej i analiza treści plików nagłówkowych i dostarczonych funkcji:
   - uwaga na przekazywanie argumentu *Glowa_wsk* np. do funkcji wstawiania na początek listy – *Głowa* jest wskaźnikiem, którego wartość jest zmieniana, jako argument musi zostać przekazany wskaźnik do *Głowa*, czyli wskaźnik do wskaźnika, typu *el_list***
2. Uruchomienie kodu, sprawdzenie poprawności działania
3. Modyfikacja plików *lista_powiazana.c i lista_powiazana.h*, tak aby wzbogacić interfejs o funkcje wstawiania elementu do środka listy (jaki musi być interfejs takiej funkcji?) i na koniec listy
4. Uzupełnienie funkcji *main* o odpowiednie wywołania – testowanie poprawności działania
5. Modyfikacja plików *lista_powiazana.c i lista_powiazana.h*, tak aby identyfikatorem listy (uchwytem dającym do niej dostęp) nie był wskaźnik do "głowy", ale wskaźnik do struktury zawierającej "głowę", np.

```
typedef struct {
 int liczba_elementow;
 wsk_el_list Glowa;   // ewentualnie: el_list* Glowa;
} lista;
```

   - inicjowanie takiej listy może być przeprowadzane funkcją *inicjuj_lista* (uzupełniającą interfejs listy – deklaracja w *lista_powiazana.h* , definicja w *lista_powiazana.c*):

```
lista* inicjuj_lista()
{
 lista* Lista_wsk;
 Lista_wsk = (lista*) malloc(sizeof(lista));
 Lista_wsk->Glowa = NULL;
 Lista_wsk->liczba_elementow = 0;
 return(Lista_wsk);
}
```

   - wprowadzenie identyfikacji poprzez wskaźnik do struktury oznacza, że w kodzie korzystającym z naszej implementacji listy powiązanej inicjowanie listy odbywa się poprzez wywołanie

```
lista* lista_wsk = inicjuj_lista();
```

     (typ *lista* jest elementem interfejsu)
   - w dowolnej funkcji korzystającej z tak zainicjowanej listy, jako pierwszy argument identyfikujący listę w funkcjach korzystających z listy, pojawiać się teraz będzie wskaźnik do struktury otrzymany z funkcji *inicjuj_lista()*, np. *drukuj_lista(lista_wsk);*
   - oznacza to, że wszystkie dotychczasowe deklaracje i definicje funkcji z *lista_powiazana.c i lista_powiazana.h* muszą zostać odpowiednio zmodyfikowane, tak żeby typem pierwszego argumentu stał się *lista**
6. Modyfikacja plików źródłowych implementacji listy powiązanej, tak aby obsługiwać listę dwukierunkową:
   - w strukturze listy przechowywany jest nie tylko wskaźnik do pierwszego elementu ("głowa"), ale także do ostatniego ("ogon"),
   - każdy element zawiera wskaźnik nie tylko do elementu następnego, ale także do poprzedniego, dzięki czemu możliwe jest przeglądanie listy nie tylko od przodu do tyłu, ale także od tyłu do przodu