

Wstęp do Programowania – Listy

Laboratorium nr 5i6 (4 x 45")

Grupa WCY21KY2S1

29.05.2023

Temat:

T5 (I). Podstawowe struktury dynamiczne - dwukierunkowe, cykliczne.

- *Struktura listy dwukierunkowej i jej modyfikacje w zależności od typów danych,*
- *Procedury dodania, wstawiania i usunięcia elementów z listy,*
- *Usuwanie listy i generacje kilku niezależnych list,*
- *Lista cykliczna i jej właściwości, skanowanie elementów listy cyklicznej,*
- *Modyfikacje elementów listy.*

T6 (I). Struktury mieszane: tablice z dowiązaniem listowymi.

- *Powiązanie tablic i list jednokierunkowych – techniki haszowania podstawowego*
-

Zad. nr 1

Wygeneruj w programie dwukierunkową listę cykliczną. Maksymalnie, lista może zawierać k-elementów. Do listy należy jednak wprowadzić n-elementów, gdzie $n > k$. Wartości n i k podaje użytkownik. Elementy listy są generowane losowo z zakresu (1,100). Elementy będą dodawane wg. zasad:

- nowe elementy są dodawane do listy na pole x w kolejności narastającej,
- po osiągnięciu maksymalnej ilości elementów – k, z listy należy usunąć powtarzające się wartości. Ewentualną różnicę $n-k$, uzupełnić wartościami losowymi.
- wartości listy po przekroczeniu ilości k-elementów, należy dodawać tak, aby tworzyły listę dowiązaną (*nlist) do już istniejących wg. zasady:
 - wartość nowego elementu x_R dołączana jest do elementu x_i tak, aby $x_i \leq x_R < x_{i+1}$. Elementy o wartościach mniejszych od najmniejszego – odrzucamy, większe od największego dołączamy do największego elementu. Elementy x_R tworzą listę dwukierunkową wraz z elementem x_i .

Wykorzystaj strukturę główną np.:

```
typedef struct element {
    int x;
    struct element * next;
    struct element * prev;
    struct element * nlist;
} elem_t;

elem_t *head;
```

Oprogramuj niżej wymienione następujące funkcjonalności:

- a) **Insert**(L, x) – funkcję, która pozwala dodać do listy L element x ,
- b) **InsertS**(L, x) – funkcję, która wstawi element tak, aby układ elementów tworzył niemalejący ciąg wartości elementów przechowywanych /dodatkowo/,
- c) **PrintAll**(L) – funkcję, która pozwala wypisać elementy listy L ,
- d) **PrintAll**(L, x) – funkcję, która pozwoli wypisać wszystkie podzielniki x na liście dowiązanej,
- e) **PrintAllA**(L) – funkcję, która wypisze wszystkie elementy listy L wraz z sąsiadami np. dla elementów listy: 5,6,7,8:

$8 \leftarrow 5 \rightarrow 6$
$5 \leftarrow 6 \rightarrow 7$
$6 \leftarrow 7 \rightarrow 8$
$7 \leftarrow 8 \rightarrow 5$

- f) **DelElm**(L, x) – funkcję, która usunie element x , z listy (wszystkie wystąpienia elementu),
- g) **DelList**(L) – funkcję, która pozwala usunąć listę z pamięci,

Uwagi dodatkowe:

- Kod programu należy umieścić w jednym pliku (włącznie z funkcjami) [pdf lub docx]. Przedstawione rozwiązanie powinno być wykonane samodzielnie. W przypadku stwierdzenia kopii wszyscy autorzy identycznych rozwiązań otrzymują ocenę ndst., co wiąże się z konieczności poprawy ćwiczenia w terminie konsultacji.
- Do implementacji proszę użyć języka C w standardzie ANSI C,
- Do ćwiczenia laboratoryjnego sporządzamy notatkę w formie elektronicznej, która jest dodawana do ćwiczenia w oddzielnym pliku w formacie {docx, pdf, odt}. W notatce zamieszczamy odpowiedzi na pytania opisowe i niezbędne obliczenia.
- Sprawdzenie pracy następuje w czasie zajęć laboratoryjnych, po zgłoszeniu przez studenta gotowości do sprawdzenia pracy.
- Gotowe rozwiązania (sprawozdanie i kod programu – dwa pliki niespakowane) przesyłamy za pomocą narzędzia MS Teams do oceny. Ocena po sprawdzeniu będzie dostępna dla każdego studenta w systemie MS Teams.