

Pracownia programowania - Wykład 14

Listy jednokieronowe - wybrane przypadki

Repozytoria

- Krzaczkowski
 - <https://github.com/pjastr/Listy>
 - <https://github.com/pjastr/ListyJednokierunkowe2023>
- Drugi zbiór:
 - <https://github.com/pjastr/zbior-zadan-c-rozw>

Lista bez głowy - wyświetlanie

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  struct element {
5      int x;
6      struct element * next;
7  };
8
9  void printListWithoutHead(struct element *list) {
10     if (list == NULL) {
11         printf("Lista jest pusta\n");
12         printf("---\n");
13         return;
14     }
15     struct element *current = list;
16     while (current != NULL) {
17         printf("%d\n", current->x);
18         current = current->next;
```

Lista z głową - wyświetlanie

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  struct element {
5      int x;
6      struct element * next;
7  };
8
9  void printListWithHead(struct element *list) {
10     if (list->next == NULL) {
11         printf("Lista jest pusta\n");
12         printf("---\n");
13         return;
14     }
15     struct element *current = list->next;
16     while (current != NULL) {
17         printf("%d\n", current->x);
18         current = current->next;
```

Lista bez głowy - tworzenie

```
1  #include <stdio.h>
2
3  struct element {
4      int x;
5      struct element * next;
6  };
7
8  struct element * createNoHead() {
9      return NULL;
10 }
11
12 int main() {
13     struct element * list = createNoHead();
14     printf("%p\n", list);
15     return 0;
16 }
```

Lista z głową - tworzenie

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  struct element {
5      int x;
6      struct element * next;
7  };
8
9  struct element * createWithHead() {
10     struct element * head = malloc(sizeof(struct element));
11     head->next = NULL;
12     return head;
13 }
14
15 int main() {
16     struct element * list = createWithHead();
17     printf("%p\n", list);
18     printf("%p\n", list->next);
```

Lista bez głowy - dodanie na początek - pojedynczy wskaźnik

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  struct element {
5      int x;
6      struct element * next;
7  };
8
9  //rozwiązanie za pomocą pojedynczych wskaźników
10 struct element * addFirst(struct element *list, int a) {
11     struct element *newElement = malloc(sizeof(struct element));
12     newElement->x = a;
13     newElement->next = list;
14     return newElement;
15 }
16
17 void printList(struct element *list) {
18     while (list != NULL) {
```


Lista bez głowy - dodanie na początek - podwójny wskaźnik

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  struct element {
5      int x;
6      struct element * next;
7  };
8
9  //rozwiązanie za pomocą podwojnego wskaźnika
10 void addFirst(struct element ** list, int a) {
11     struct element *newElement = malloc(sizeof(struct element));
12     newElement->x = a;
13     newElement->next = *list;
14     *list = newElement;
15 }
16
17 void printList(struct element *list) {
18     while (list != NULL) {
```

Lista z głową - dodanie na początek

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  struct element {
5      int x;
6      struct element * next;
7  };
8
9  void addFirst(struct element *list, int a) {
10     struct element *newElement = malloc(sizeof(struct element));
11     newElement->x = a;
12     newElement->next = list->next;
13     list->next = newElement;
14 }
15
16 void printList(struct element *list) {
17     struct element *tmp = list->next;
18     while (tmp != NULL) {
```

Lista bez głowy - dodanie na koniec - pojedynczy wskaźnik

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  struct element {
5      int x;
6      struct element * next;
7  };
8
9  struct element * addLast(struct element *list, int a) {
10     struct element *newElement = malloc(sizeof(struct element));
11     newElement->x = a;
12     newElement->next = NULL;
13     if (list == NULL) {
14         return newElement;
15     }
16     struct element *current = list;
17     while (current->next != NULL) {
18         current = current->next;
```

Lista bez głowy - dodanie na koniec - podwójny wskaźnik

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  struct element {
5      int x;
6      struct element * next;
7  };
8
9  void addLast(struct element ** list, int a) {
10     struct element *newElement = malloc(sizeof(struct element));
11     newElement->x = a;
12     newElement->next = NULL;
13     if (*list == NULL) {
14         *list = newElement;
15         return;
16     }
17     struct element *current = *list;
18     while (current->next != NULL) {
```

Lista z głową - dodanie na koniec

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  struct element {
5      int x;
6      struct element * next;
7  };
8
9  void addLast(struct element *list, int a) {
10     struct element *current = list;
11     while (current->next != NULL) {
12         current = current->next;
13     }
14     current->next = malloc(sizeof(struct element));
15     current->next->x = a;
16     current->next->next = NULL;
17 }
18
```

Lista bez głowy - usunięcie całej listy

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  struct element {
5      int x;
6      struct element * next;
7  };
8
9  struct element * removeList(struct element * list) {
10     struct element * current = list;
11     struct element * next;
12     while (current != NULL) {
13         next = current->next;
14         free(current);
15         current = next;
16     }
17     return NULL;
18 }
```

Lista z głową - usunięcie całej listy

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  struct element {
5      int x;
6      struct element * next;
7  };
8
9  void removeList(struct element * list){
10     struct element * current = list->next;
11     struct element * next;
12     while(current != NULL){
13         next = current->next;
14         free(current);
15         current = next;
16     }
17     list->next = NULL;
18 }
```

Lista bez głowy - usunięcie początkowego elementu

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  struct element {
5      int x;
6      struct element * next;
7  };
8
9  struct element * removeFirst(struct element * list) {
10     if (list == NULL) {
11         return NULL;
12     }
13     struct element * newHead = list->next;
14     free(list);
15     return newHead;
16 }
17
18 void printList(struct element * list) {
```


Lista z głową - usunięcie początkowego elementu

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  struct element {
5      int x;
6      struct element * next;
7  };
8
9  void removeFirst(struct element *list) {
10     if (list->next == NULL) {
11         return;
12     }
13     struct element *temp = list->next;
14     list->next = list->next->next;
15     free(temp);
16 }
17
18 void printList(struct element *list) {
```

Lista bez głowy - usunięcie ostatniego elementu

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  struct element {
5      int x;
6      struct element * next;
7  };
8
9  struct element * deleteLast(struct element *list) {
10     if (list == NULL) {
11         return NULL;
12     }
13     if (list->next == NULL) {
14         free(list);
15         return NULL;
16     }
17     struct element *current = list;
18     while (current->next->next != NULL) {
```

Lista z głową - usunięcie ostatniego elementu

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  struct element {
5      int x;
6      struct element * next;
7  };
8
9  void removeLast(struct element * list) {
10     if (list->next == NULL) {
11         return;
12     }
13     struct element * prev = list;
14     struct element * current = list->next;
15     while (current->next != NULL) {
16         prev = current;
17         current = current->next;
18     }
```

Lista bez głowy - usunięcie przedostatniego elementu

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  struct element {
5      int x;
6      struct element * next;
7  };
8
9  struct element * deleteBeforeLast(struct element *list) {
10     if (list == NULL || list->next == NULL) {
11         return list;
12     }
13     struct element *prev = NULL;
14     struct element *current = list;
15     struct element *next = list->next;
16     while (next->next != NULL) {
17         prev = current;
18         current = next;
```

Lista z głową - usunięcie przedostatniego elementu

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 struct element {
5     int x;
6     struct element * next;
7 };
8
9 void removeBeforeLast(struct element * list) {
10     if (list->next == NULL || list->next->next == NULL) {
11         return;
12     }
13     struct element * prev = list;
14     struct element * current = list->next;
15     struct element * next = current->next;
16     while (next->next != NULL) {
17         prev = current;
18         current = next;
```