Ad soyad : **Yasin KARAÇAM**

**Veri Yapıları - I Ödev 1**

**Soru :**

1. Tek bağlı ve çift bağlı bir listenin Palindrom (Palindrom, tersten okunuşu da aynı olan cümle, sözcük ve sayılara denilmektedir) olup olmadığını kontrol eden bir C program yazınız.
2. Her iki program içinde yazdığınız kodları editörden cevap kâğıdına kopyalayarak aktarınız.
3. Programlarınızda dört örnek için (ikisi Palindrom olan ve diğer ikisi olmayan) program çıktılarını da cevap kâğıdına kopyalayarak aynen veriniz.

**Çözüm :**

**Palindrome mu Değil mi ?**

Singly Linked List :

#include <stdio.h>

#include <stdlib.h>

#include <stdbool.h>

#include <string.h>

struct Node {

char data;

struct Node\* next;

};

bool isPalindrome(struct Node\* head) {

struct Node\* temp1 = head;

struct Node\* temp2 = NULL;

struct Node\* reverse = NULL;

while (temp1 != NULL) {

struct Node\* new\_node = (struct Node\*)malloc(sizeof(struct Node));

new\_node->data = temp1->data;

new\_node->next = reverse;

reverse = new\_node;

temp1 = temp1->next;

}

temp1 = head;

temp2 = reverse;

while (temp1 != NULL && temp2 != NULL) {

if (temp1->data != temp2->data) {

return false;

}

temp1 = temp1->next;

temp2 = temp2->next;

}

return true;

}

void push(struct Node\*\* head, char new\_data) {

struct Node\* new\_node = (struct Node\*)malloc(sizeof(struct Node));

new\_node->data = new\_data;

new\_node->next = \*head;

\*head = new\_node;

}

void printList(struct Node\* head) {

struct Node\* temp = head;

while (temp != NULL) {

printf("%c", temp->data);

if (temp->next != NULL) {

printf(" -> ");

}

temp = temp->next;

}

}

int main() {

struct Node\* head = NULL;

char input[100];

printf("Bir kelime girin: ");

fgets(input, sizeof(input), stdin);

input[strcspn(input, "\n")] = 0;

for (int i = strlen(input) - 1; i >= 0; i--) {

push(&head, input[i]);

}

printf("Tek Bağlı Liste: ");

printList(head);

if (isPalindrome(head))

printf("\nListe bir palindromdur.\n");

else

printf("\nListe bir palindrom değildir.\n");

return 0;

}

Doubly Linked List :

#include <stdio.h>

#include <stdlib.h>

#include <stdbool.h>

#include <string.h>

struct Node {

char data;

struct Node\* next;

};

bool isPalindrome(struct Node\* head) {

struct Node\* temp1 = head;

struct Node\* temp2 = NULL;

struct Node\* reverse = NULL;

while (temp1 != NULL) {

struct Node\* new\_node = (struct Node\*)malloc(sizeof(struct Node));

new\_node->data = temp1->data;

new\_node->next = reverse;

reverse = new\_node;

temp1 = temp1->next;

}

temp1 = head;

temp2 = reverse;

while (temp1 != NULL && temp2 != NULL) {

if (temp1->data != temp2->data) {

return false;

}

temp1 = temp1->next;

temp2 = temp2->next;

}

return true;

}

void push(struct Node\*\* head, char new\_data) {

struct Node\* new\_node = (struct Node\*)malloc(sizeof(struct Node));

new\_node->data = new\_data;

new\_node->next = \*head;

\*head = new\_node;

}

void printList(struct Node\* head) {

struct Node\* temp = head;

while (temp != NULL) {

printf("%c", temp->data);

if (temp->next != NULL) {

printf(" -> ");

}

temp = temp->next;

}

}

int main() {

struct Node\* head = NULL;

char input[100];

printf("Bir kelime girin: ");

fgets(input, sizeof(input), stdin);

input[strcspn(input, "\n")] = 0;

for (int i = strlen(input) - 1; i >= 0; i--) {

push(&head, input[i]);

}

printf("Tek Bağlı Liste: ");

printList(head);

if (isPalindrome(head))

printf("\nListe bir palindromdur.\n");

else

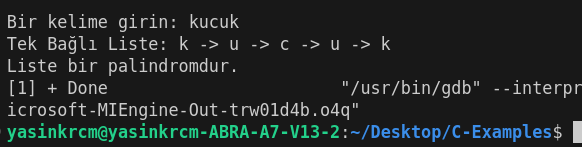
printf("\nListe bir palindrom değildir.\n");

return 0;

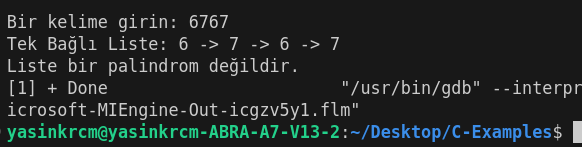
}

Çıktı Ekran görüntüleri

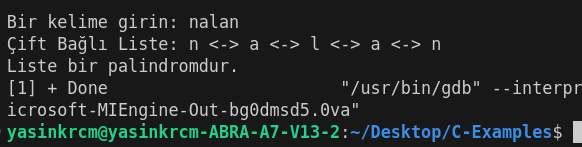
1. Tek bağlı Liste :
2. Doğru örnek :



1. Yanlış örnek :



1. Çift bağlı liste :
2. Doğru örnek:



1. Yanlış örnek:

