Prefix İfadelerin Kuyruk Yapısı İle Hesaplanması

Zehra Aktürk

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

**//Bağlı liste oluşturmak için bir struct tanımlanır.**

struct node {

int data;

struct node\* next;

};

typedef struct node Node;

**// Düğüm oluşturduğumuz fonksiyon bu fonksiyondur.**

Node\* CreateNode(int data) {

Node\* node = (Node\*)malloc(sizeof(Node));

node->data = data;

node->next = NULL;

return node;

}

**//Kuyruğa ekleme fonksiyonu head ın değeri değiştiği için Node \*\*(çift pointer ) kullanılmıştır.**

void KuyrugaEkle(Node\*\* head, int data) {

Node\* node = CreateNode(data);

node->next = \*head;

\*head = node;

}

**//Kuyruktan alma fonksiyonu yine head değişiyor.**

int KuyruktanAl(Node\*\* head) {

if (\*head == NULL) {

return -1;

}

int data = (\*head)->data;

Node\* temp = \*head;

\*head = (\*head)->next;

free(temp);

return data;

}

**//Operatör olup olmadığını kontrol eden fonksiyon**

int IsOperator(char c) {

return c == '+' || c == '-' || c == '\*' || c == '/';

}

**//Hesaplamaya karar verilen fonksiyon**

int EvaluatePrefixExpression(char\* expression) {

Node\* head = NULL;

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

if (IsOperator(expression[i])) {

int operand1 = KuyruktanAl(&head);

int operand2 = KuyruktanAl(&head);

// Operat r n sonucunu hesaplar

int result = 0;

switch (expression[i]) {

case '+':

result = operand1 + operand2;

break;

case '-':

result = operand1 - operand2;

break;

case '\*':

result = operand1 \* operand2;

break;

case '/':

result = operand1 / operand2;

break;

}

KuyrugaEkle(&head, result);

} else {

KuyrugaEkle(&head, expression[i] - '0');

}

}

if (head != NULL) {

return head->data;

}

return -1;

}

int main() {

char expression[] = "+\*9+28\*+4863";

int result = EvaluatePrefixExpression(expression);

printf("Ifadenin degeri: %d\n", result);

return 0;

}