#include <stdio.h>

#include <stdlib.h>

struct node {

int value;

struct node \*next;

};

typedef struct node\* NODE;

NODE getnode() {

NODE new\_node = (NODE)malloc(sizeof(struct node));

if (new\_node == NULL) {

printf("Memory allocation failed.\n");

exit(1);

}

return new\_node;

}

void display(NODE first) {

NODE temp;

if (first == NULL) {

printf("Linked list is empty\n");

return;

}

temp = first;

while (temp != NULL) {

printf("%d -> ", temp->value);

temp = temp->next;

}

printf("NULL\n"); // To indicate the end of the list

}

NODE insert\_beg(int item, NODE first) {

NODE new = getnode();

new->value = item;

new->next = first;

return new; // Return the new head of the list

}

NODE delete\_end(NODE first) {

if (first == NULL) {

printf("Linked list is empty\n");

return NULL;

}

if (first->next == NULL) {

free(first);

return NULL; // If there's only one node

}

NODE prev = NULL, last = first;

while (last->next != NULL) {

prev = last;

last = last->next;

}

prev->next = NULL;

free(last);

return first; // Return the updated list

}

int main() {

int choice, item;

NODE first = NULL; // Initialize the first pointer to NULL

NODE item\_del;

while (1) {

printf("Enter your choice:\n1. Insert\n2. Delete\n3. Display\n");

scanf("%d", &choice);

switch (choice) {

case 1:

printf("Enter item to insert at the beginning:\n");

scanf("%d", &item);

first = insert\_beg(item, first); // Update the first pointer

break;

case 2:

first = delete\_end(first); // Update the first pointer

break;

case 3:

printf("The linked list is being displayed:\n");

display(first);

break;

default:

printf("Wrong choice\n");

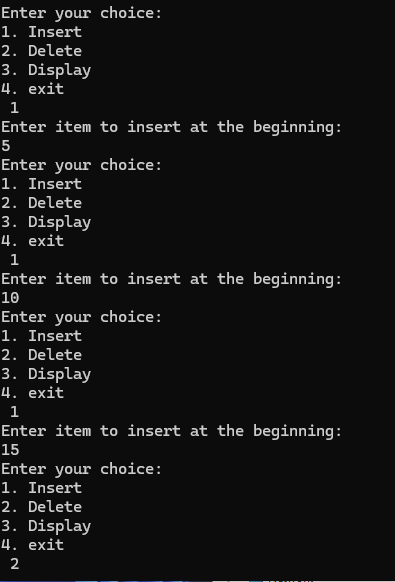
exit(0);

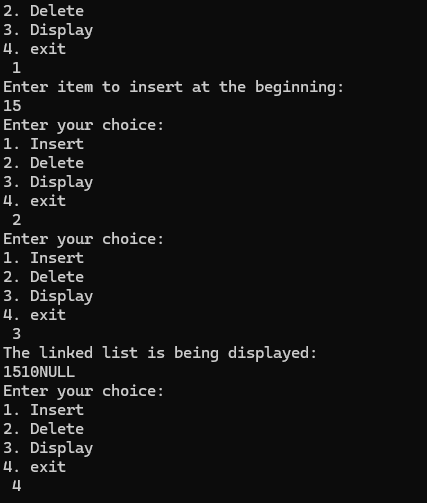
}

}

return 0;

}





#include <stdio.h>

#include <stdlib.h>

struct node {

int value;

struct node \*next;

};

typedef struct node\* NODE;

NODE getnode() {

NODE new\_node = (NODE)malloc(sizeof(struct node));

if (new\_node == NULL) {

printf("Memory allocation failed.\n");

exit(1);

}

return new\_node;

}

void display(NODE first) {

if (first == NULL) {

printf("Linked list is empty\n");

return;

}

NODE temp = first;

while (temp != NULL) {

printf("%d ", temp->value);

temp = temp->next;

}

printf("\n");

}

NODE insert\_beg(int item, NODE first) {

NODE new = getnode();

new->value = item;

new->next = first;

return new;

}

NODE delete\_first(NODE first) {

if (first == NULL) {

printf("Linked list is empty, nothing to delete\n");

return NULL;

}

NODE temp = first;

first = first->next;

printf("Item deleted: %d\n", temp->value);

free(temp);

return first;

}

int main() {

int choice, item;

NODE first = NULL;

while (1) {

printf("Enter your choice: \n1. Push\n2. Pop\n3. Display\n4. Exit\n");

scanf("%d", &choice);

switch (choice) {

case 1:

printf("Enter item to push: ");

scanf("%d", &item);

first = insert\_beg(item, first);

break;

case 2:

first = delete\_first(first);

break;

case 3:

printf("The stack is being displayed: ");

display(first);

break;

case 4:

exit(0);

default:

printf("Invalid choice\n");

}

}

return 0;

}

