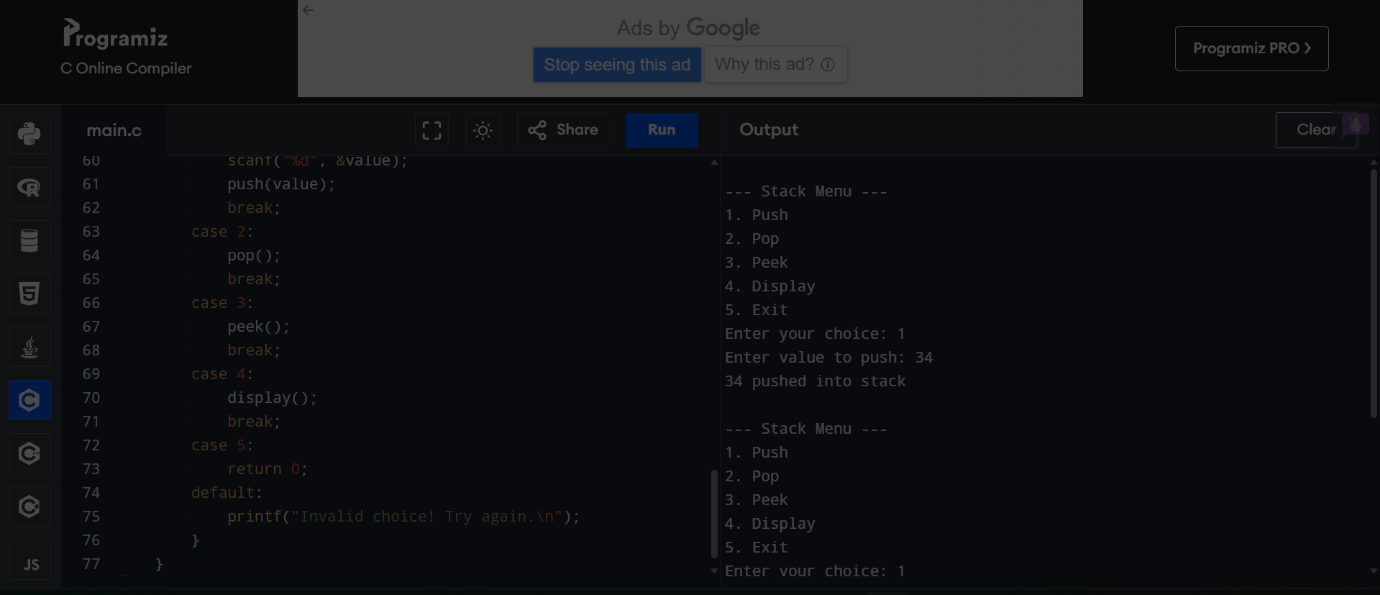
Implementation of stack using linked list



#include <stdio.h>

#include <stdlib.h>

struct Node {

int data;

struct Node\* next;

};

struct Node\* top = NULL;

void push(int value) {

struct Node\* newNode = (struct Node\*)malloc(sizeof(struct Node));

if (newNode == NULL) {

printf("Stack Overflow! Memory not available.\n");

return;

}

newNode->data = value;

newNode->next = top;

top = newNode;

printf("%d pushed into stack\n", value);

}

void pop() {

if (top == NULL) {

printf("Stack Underflow! Nothing to pop.\n");

return;

}

struct Node\* temp = top;

printf("%d popped from stack\n", top->data);

top = top->next;

free(temp);

}

void peek() {

if (top == NULL) {

printf("Stack is empty\n");

} else {

printf("Top element: %d\n", top->data);

}

}

void display() {

if (top == NULL) {

printf("Stack is empty\n");

return;

}

struct Node\* temp = top;

printf("Stack elements: ");

while (temp != NULL) {

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

temp = temp->next;

}

printf("\n");

}

int main() {

int choice, value;

while (1) {

printf("\n--- Stack Menu (Linked List Implementation) ---\n");

printf("1. Push\n2. Pop\n3. Peek\n4. Display\n5. Exit\n");

printf("Enter your choice: ");

scanf("%d", &choice);

switch (choice) {

case 1:

printf("Enter value to push: ");

scanf("%d", &value);

push(value);

break;

case 2:

pop();

break;

case 3:

peek();

break;

case 4:

display();

break;

case 5:

return 0;

default:

printf("Invalid choice! Try again.\n");

}

}

}