```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#define MAX 100
//stack structure
typedef struct{
   char data[MAX] [MAX];
   int top;
} Stack;

//Function to initialize stack
void initStack(Stack*s){
   s->top=-1;
}
//Push to stack
void push(Stack*s,const char*str){
   if (s->top<MAX-1){
      strcpy(s->data[++s->top],str);
   }else{
      printf("Stack Overflow!\n");
   }

}
//Pop from stack
char*pop(Stack*s){
   if(s->top>=0)
     return s->data[s->top--];
   else
     return NULL;
}
//check if stack is empty
int isEmpty(Stack*s){
   return s->top == -1;

}
int main(){
   Stack undoStack,redoStack;
   initStack(&undoStack);
   initStack(&redoStack);

   char currentTex[MAX]="";
   char action[MAX];
   int choice;
   while(1){
      printf("\n----Text editor Simulation----\n");
      printf("Current Tex:\"%s\"\n",currentTex);
      printf("1.Add Text\n2.Undo\n3.Redo\n4.Exit\n");
      printf("Enter your choice:");
```

```
        scanf("%d",&choice);
        getchar(); //consume newline
        switch(choice){
           case 1:
            printf("Enter text to add:");
            fgets(action,MAX,stdin);
            action[strcspn(action,"\n")]=0; //remove newline

            //save current state to undo stack
            push(&undoStack,currentTex);

            //clear redo stack
            initStack(&redoStack);

            //append new text
            strcat(currentTex,action);
            break;
           case 2: //redo
            if(!isEmpty(&undoStack)){
              push(&redoStack,currentTex);
              strcpy(currentTex,pop(&undoStack));

            }else{
            printf("Nothing to undo!\n");
            }
            break;
           case 3:
            if(!isEmpty(&redoStack)){
              push(&undoStack,currentTex);
              strcpy(currentTex,pop(&redoStack));
            }else{
              printf("Nothing to redo!\n");
            }
            break;
           case 4:
            exit(0);
           default:
            printf("Invalid choice!\n");

        }

    }
    return 0;
}
```