import java.util.\*;

interface Stack {

    void push(String item);

    String pop();

    String peek();

    boolean isEmpty();

}

class ArrayStack implements Stack {

    private String[] elements = new String[10];

    private int top = -1;

    public void push(String item) {

        if (top == elements.length - 1) {

            System.out.println("stack is full! cannot push");

            return;

        }

        elements[++top] = item;

    }

    public String pop() {

        if (isEmpty()) {

            System.out.println("stack is empty");

            return null;

        }

        return elements[top--];

    }

    public String peek() {

        if (isEmpty()) return null;

        return elements[top];

    }

    public boolean isEmpty() {

        return top == -1;

    }

}

public class Texteditor {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        ArrayStack undoStack = new ArrayStack();

        while (true) {

            System.out.println(" text editor ");

            System.out.println("1. type text");

            System.out.println("2. undo");

            System.out.println("3. view current text");

            System.out.println("4. exit");

            System.out.print("enter choice: ");

            int choice = sc.nextInt();

            sc.nextLine();

            switch (choice) {

                case 1:

                    System.out.println("enter text");

                    String text = sc.nextLine();

                    undoStack.push(text);

                    break;

                case 2:

                    String undone = undoStack.pop();

                    if (undone != null) {

                        System.out.println("undo: " + undone);

                    }

                    break;

                case 3:

                    String current = undoStack.peek();

                    System.out.println("current text: " + (current != null ? current : "no text"));

                    break;

                case 4:

                    System.out.println("exiting");

                    sc.close();

                    return;

                default:

                    System.out.println("invalid");

            }

        }

    }

}