Ankit is a 5-year-old kid, who is playing in a boardroom with a basket and balls. Each ball is having numbered on it like 1, 2, 3, …, 9. Ankit's aunt asked him to put all boll inside the basket. Here, the scenario is that basket is long enough to hold all balls but squeezed in width and can hold only one boll at a time. When the next ball is inserted, then that ball can lie on top of the old boll like this way.

- Push ball numbered as 1 inside the basket
- Push ball numbered as 8 inside the basket
- Push ball numbered as 9 inside the basket
- Push ball numbered as 7 inside the basket
- Push ball numbered as 2 inside the basket
- Pop ball from the basket
- Pop ball from the basket
- Push ball numbered as 3 inside the basket

Input Format

1 8 9 7 2 3

Output Format

1 8 9 3

Constraints

 size should not exceed 9

Code:

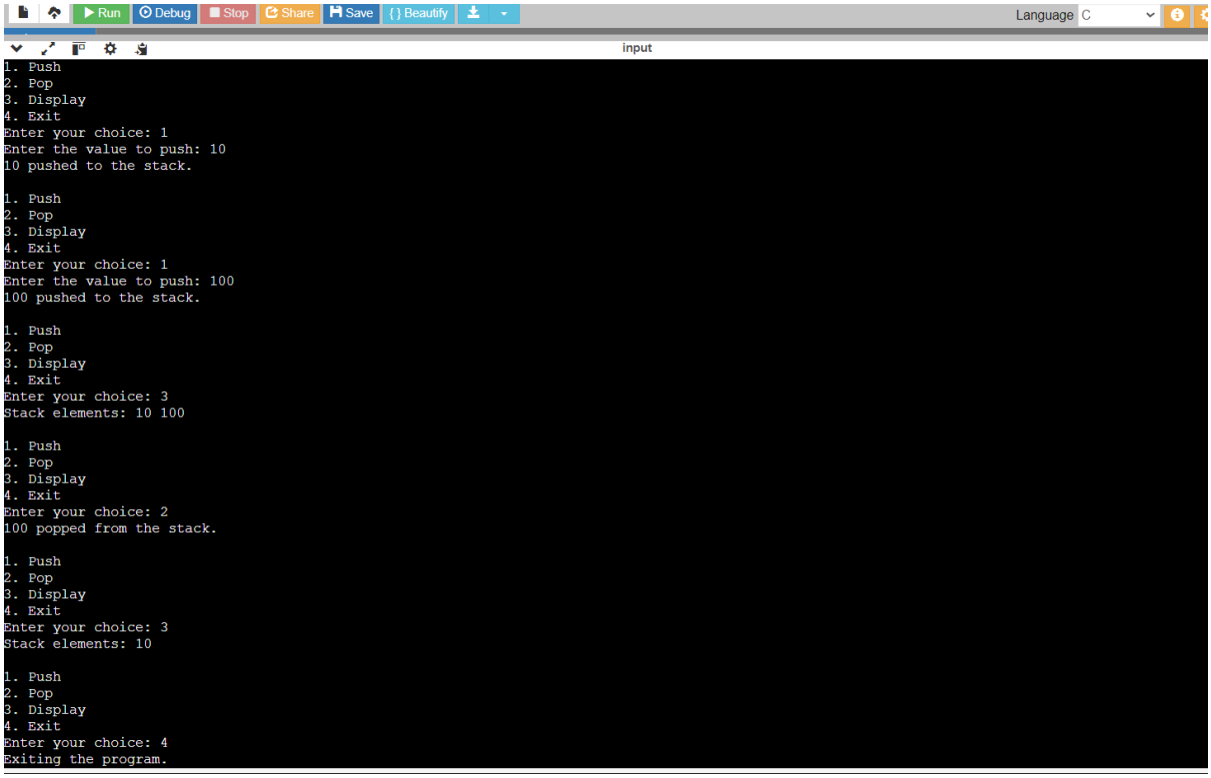
```
#include <stdio.h>
#include <stdbool.h>

#define MAX_SIZE 100

typedef struct {
    int arr[MAX_SIZE];
    int top;
```

```
} Stack;

void initStack(Stack *stack) {
    stack->top = -1;
}

bool isFull(Stack *stack) {
    return stack->top == MAX_SIZE - 1;
}

bool isEmpty(Stack *stack) {
    return stack->top == -1;
}

void push(Stack *stack, int value) {
    if (isFull(stack)) {
        printf("Stack Overflow! Cannot push element.\n");
        return;
    }
    stack->arr[++stack->top] = value;
    printf("%d pushed to the stack.\n", value);
}

void pop(Stack *stack) {
    if (isEmpty(stack)) {
        printf("Stack Underflow! Cannot pop element.\n");
        return;
    }
    int value = stack->arr[stack->top--];
    printf("%d popped from the stack.\n", value);
}
```

```
void display(Stack *stack) {
    if (isEmpty(stack)) {
        printf("Stack is empty.\n");
        return;
    }
    printf("Stack elements: ");
    for (int i = 0; i <= stack->top; i++) {
        printf("%d ", stack->arr[i]);
    }
    printf("\n");
}

int main() {
    Stack stack;
    int choice, value;

    initStack(&stack);

    do {
        printf("\n1. Push\n");
        printf("2. Pop\n");
        printf("3. Display\n");
        printf("4. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                printf("Enter the value to push: ");
                scanf("%d", &value);
```

```
            push(&stack, value);
            break;
        case 2:
            pop(&stack);
            break;
        case 3:
            display(&stack);
            break;
        case 4:
            printf("Exiting the program.\n");
            break;
        default:
            printf("Invalid choice! Please try again.\n");
      }
   } while (choice != 4);

   return 0;
}
```

Output:


Run
Debug
Stop
Share
Save
{ } Beautify
Language C
input
1. Push
2. Pop
3. Display
4. Exit
Enter your choice: 1
Enter the value to push: 10
10 pushed to the stack.
1. Push
2. Pop
3. Display
4. Exit
Enter your choice: 1
Enter the value to push: 100
100 pushed to the stack.
1. Push
2. Pop
3. Display
4. Exit
Enter your choice: 3
Stack elements: 10 100
1. Push
2. Pop
3. Display
4. Exit
Enter your choice: 2
100 popped from the stack.
1. Push
2. Pop
3. Display
4. Exit
Enter your choice: 3
Stack elements: 10
1. Push
2. Pop
3. Display
4. Exit
Enter your choice: 4
Exiting the program.