

DATE: _____ PAGE: _____

```

int pop (struct stack *s) {
    if (isEmpty(s)) {
        printf("stack underflow! cannot pop\n");
        return -1;
    } else {
        int value = s->arr[(s->top)--];
        return value;
    }
}

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

int main () {
    struct stack s;
    initstack (&s) // and symbols

    int choice, value;
    while (1) {
        printf("\n stack operation:\n");
        printf("1. 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 value to push:");
        scanf("%d", &value);
        push(&s, value);
        break;
    case 2:
        value = pop(&s);
        if (value != -1) {
            printf("popped value: %d\n", value);
        }
        break;
    case 3:
        display(&s);
        break;
    case 4:
        printf("Existing... \n");
        exit(0);
    default:
        printf("Invalid choice.\n");
}
return 0;
}

```

Output

Stack operations

- 1) push
- 2) pop
3. display
4. Exit

Enter your choice: 1

enter your value to push: 3

3 pushed to stack



Enter your choice: 2
Enter value to push: 4
u pushed its stack

Enter your choice 2
popped value: 4

Enter your choice: 5
Invalid choice

Enter your choice: 3
stack elements: 3

Enter your choice: 4
Exiting...

29/7



DATE: 29/09/2025

- ① Write a program to simulate the working of stack using an array with the following
a) push b) pop c) display
The program should print appropriate messages for stack overflow, stack underflow

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

```
#include <stdlib.h>
```

```
#define Max 5
```

```
struct stack {
```

```
    int arr[Max];
```

```
    int top;
```

```
};
```

```
void initstack(struct stack *s) {
```

```
    s->top = -1;
```

```
void push(struct stack *s, int value) {
```

```
    if (is
```

```
void initstack
```

```
int isfull(struct stack *s) {
```

```
    return s->top == Max-1;
```

```
}
```

```
int isEmpty(struct stack *s) {
```

```
    return s->top == -1;
```

```
void push(struct stack *s, int value) {
```

```
    if (isfull(s)) {
```

```
        printf("Stack overflow! cannot push", value);
```

```
    } else {
```

```
        s->arr[++(s->top)] = value;
```

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



Shot on OnePlus

chE Reddy

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  #define MAX 5
5
6  struct Stack {
7      int arr[MAX];
8      int top;
9  };
10
11
12  void initStack(struct Stack *s) {
13      s->top = -1;
14  }
15
16
17  int isFull(struct Stack *s) {
18      return s->top == MAX - 1;
19  }
20
21
22  int isEmpty(struct Stack *s) {
23      return s->top == -1;
24  }
25
26
27  void push(struct Stack *s, int value) {
28      if (isFull(s)) {
29          printf("Stack Overflow! Cannot push %d\n", value);
30      } else {
31          s->arr[++(s->top)] = value;
32          printf("%d pushed to stack\n", value);
33      }
34  }
35
36
37  int pop(struct Stack *s) {
38      if (isEmpty(s)) {
39          printf("Stack Underflow! Cannot pop\n");
40          return -1;

```

```
        return -1;
    } else {
        int value = s->arr[(s->top)--];
        return value;
    }
}

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

int main() {
    struct Stack s;
    initStack(&s);

    int choice, value;

    while (1) {
        printf("\nStack Operations:\n");
        printf("1. 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 value to push: ");
                scanf("%d", &value);
                push(&s, value);
```

```

struct Stack s;
initStack(&s);

int choice, value;

while (1) {
    printf("\nStack Operations:\n");
    printf("1. 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 value to push: ");
            scanf("%d", &value);
            push(&s, value);
            break;
        case 2:
            value = pop(&s);
            if (value != -1) {
                printf("Popped value: %d\n", value);
            }
            break;
        case 3:
            display(&s);
            break;
        case 4:
            printf("Exiting...\n");
            exit(0);
        default:
            printf("Invalid choice. Please try again.\n");
    }
}

return 0;
}

```



```
D:\chehan\DP\stack2.exe
Stack Operations:
1. Push
2. Pop
3. Display
4. Exit
Enter your choice: 1
Enter value to push: 3
3 pushed to stack

Stack Operations:
1. Push
2. Pop
3. Display
4. Exit
Enter your choice: 1
Enter value to push: 4
4 pushed to stack

Stack Operations:
1. Push
2. Pop
3. Display
4. Exit
Enter your choice: 2
Popped value: 4

Stack Operations:
1. Push
2. Pop
3. Display
4. Exit
Enter your choice: 5
Invalid choice. Please try again.

Stack Operations:
1. Push
2. Pop
3. Display
4. Exit
Enter your choice: 3
Stack elements: 3

Stack Operations:
1. Push
2. Pop
3. Display
4. Exit
Enter your choice: 4
Exiting...

Process returned 0 (0x0)   execution time : 171.534 s
Press any key to continue.
```