

1.main.c

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

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

```
#include <string.h>
```

```
#include "header.h"
```

```
int main(){
```

```
    char exp[100];
```

```
    printf("Enter a string: ");
```

```
    scanf("%[^\n]s", exp);
```

```
    reverseString(exp);
```

```
    printf("Reversed string: %s\n", exp);
```

```
    return 0;
```

```
}
```

2.header.h

```
typedef struct {
```

```
    int top;
```

```
    char *arr;
```

```
} Stack;
```

```
void init(Stack *s, int size);
```

```
int isFull(Stack *s);
```

```
int isEmpty(Stack *s);
```

```
void push(Stack *s, int value);
```

```
char pop(Stack *s);
```

```
char peek(Stack *s);
```

```
void reverseString(char exp[]);
```

3.logic.c

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

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

```
#include <string.h>
```

```
#include "header.h"
```

```
void init(Stack *s, int size) {
```

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

```
    s->arr = (char *)malloc(sizeof(char) * size);
```

```
}
```

```
int isFull(Stack *s){
```

```
    return s -> top == 99;
```

```
}
```

```
int isEmpty(Stack *s) {
```

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

```
}
```

```
void push(Stack *s, int value) {
```

```
    if (isFull(s)) {
```

```
        printf("Stack overflow\n");
```

```
        return;
```

```
    }
```

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

```
}
```

```
char pop(Stack *s) {
```

```
    if (isEmpty(s)) {
```

```
        printf("Stack underflow\n");
```

```
    }
```

```
    return s -> arr[s->top--];
```

```
}
```

```
char peek(Stack *s) {
```

```
    if (isEmpty(s)) {
```

```
        return 0;
```

```

    }

    return s->arr[s->top];
}

void reverseString(char exp[]){
    int len = strlen(exp);

    Stack s;

    init(&s, len);

    int i;

    for(i = 0; i < len; i++){
        push(&s, exp[i]);
    }

    for(i = 0; i < len; i++){
        exp[i] = pop(&s);
    }
}

```

Output:

```

tanis@Tanishq MINGW64 /d/COEP/DSA/Serious/Assignment3/reverseString
● $ gcc -Wall main.c logic.c

tanis@Tanishq MINGW64 /d/COEP/DSA/Serious/Assignment3/reverseString
● $ ./a
Enter a string: Data Structures
Reversed string: serutcurtS ataD

tanis@Tanishq MINGW64 /d/COEP/DSA/Serious/Assignment3/reverseString
● $ ./a
Enter a string: DSA
Reversed string: ASD

tanis@Tanishq MINGW64 /d/COEP/DSA/Serious/Assignment3/reverseString
● $ ./a
Enter a string: DSA PROJECT
Reversed string: TCEJORP ASD

tanis@Tanishq MINGW64 /d/COEP/DSA/Serious/Assignment3/reverseString
● $ ./a
Enter a string: step on no pets
Reversed string: step on no pets

```