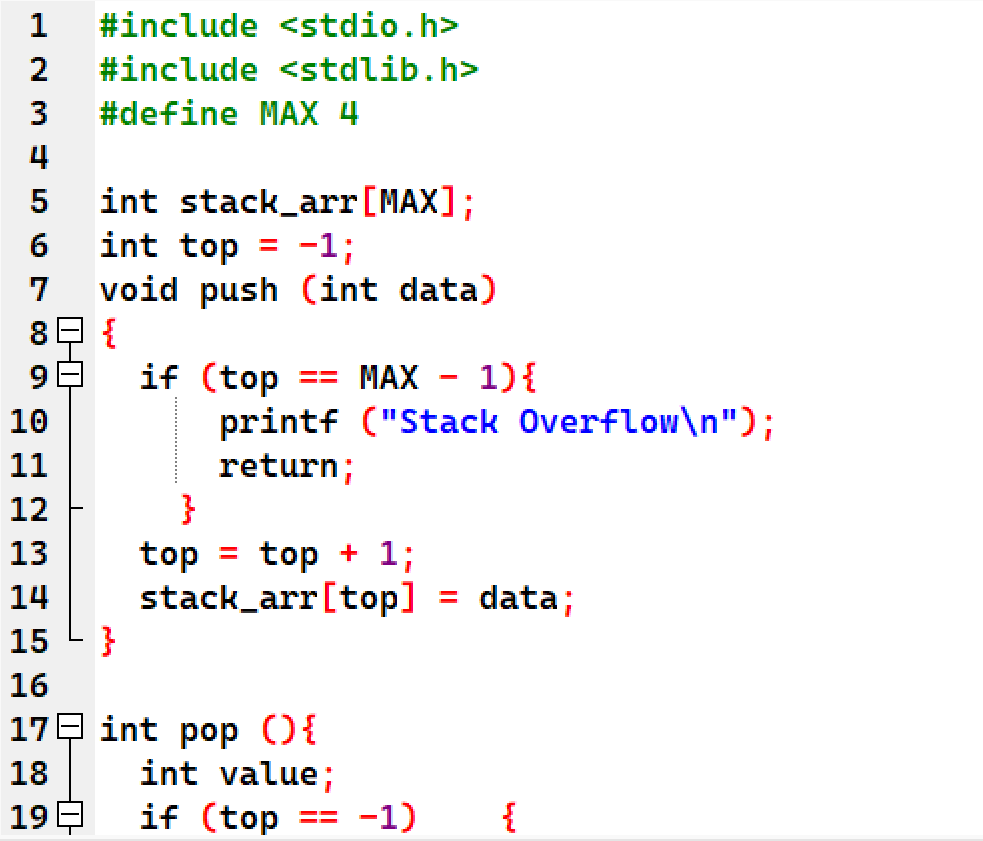
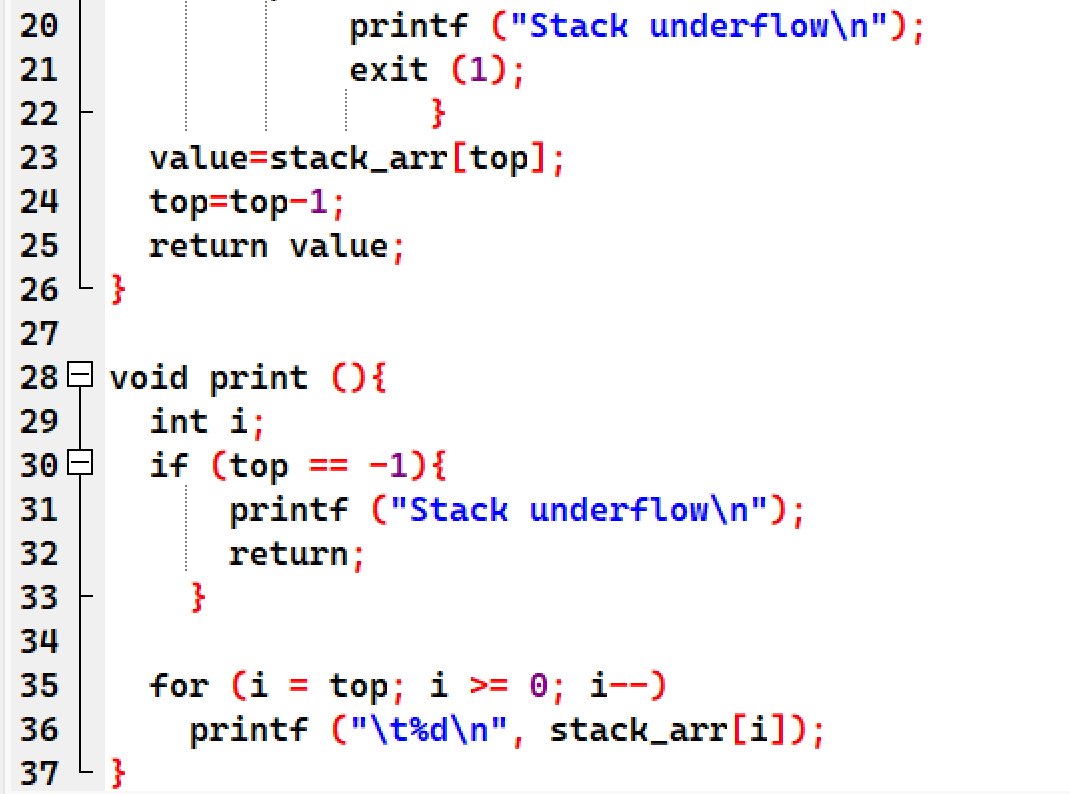
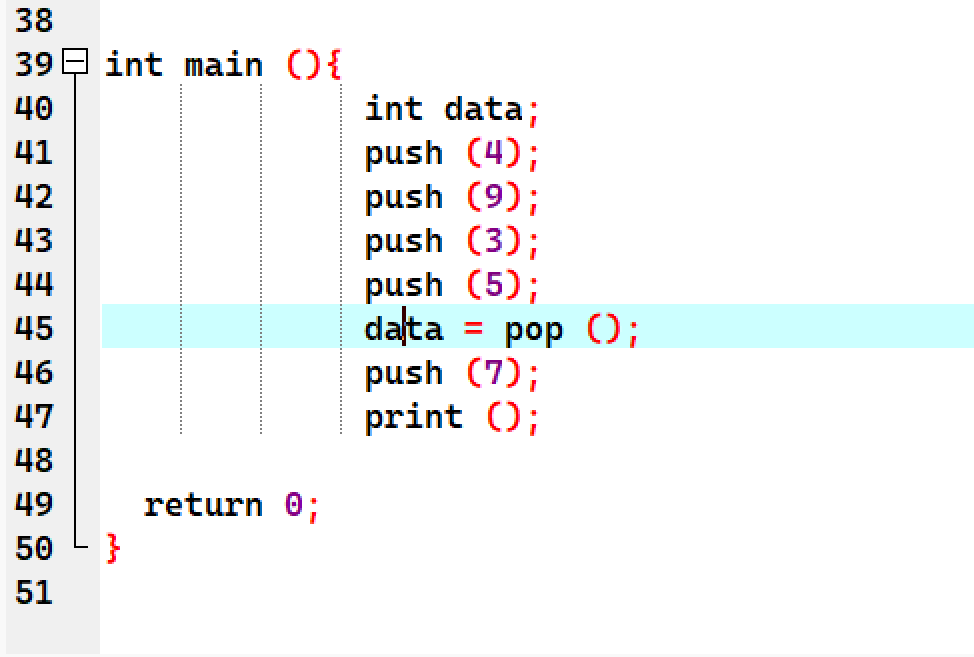
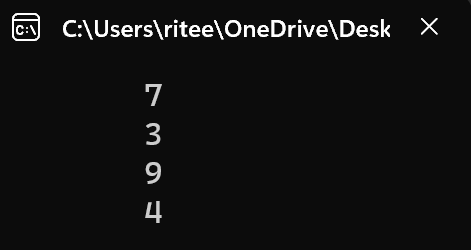
**1)**







**2)**#include <stdio.h>

#include <stdlib.h>

char stack[100];

int top = -1;

void push(char item) {

if (top >= 99) {

printf("Stack overflow\n");

exit(1);

}

stack[++top] = item;

}

char pop() {

if (top == -1) {

printf("Stack underflow\n");

exit(1);

}

return stack[top--];

}

int precedence(char ch) {

switch (ch) {

case '^':

return 3;

case '\*':

case '/':

return 2;

case '+':

case '-':

return 1;

default:

return 0;

}

}

void infixToPostfix(char \*infix, char \*postfix) {

int i, j;

char item, x;

push('(');

strcat(infix, ")");

i = 0;

j = 0;

item = infix[i];

while (item != '\0') {

if (isalnum(item)) {

postfix[j++] = item;

} else if (item == '(') {

push(item);

} else if (item == ')') {

x = pop();

while (x != '(') {

postfix[j++] = x;

x = pop();

}

} else {

while (precedence(stack[top]) >= precedence(item) && top > -1) {

x = pop();

postfix[j++] = x;

}

push(item);

}

i++;

item = infix[i];

}

postfix[j] = '\0';

}

int main() {

char infix[100], postfix[100];

printf("Enter infix expression: ");

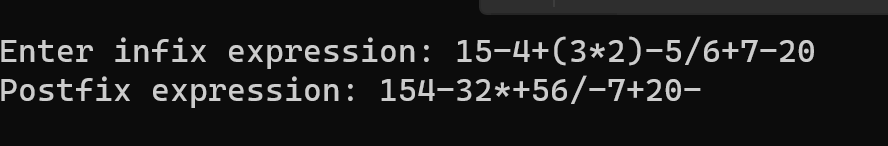
gets(infix);

infixToPostfix(infix, postfix);

printf("Postfix expression: %s\n", postfix);

return 0;

}



**3)#include <stdio.h>**

**#include <stdlib.h>**

**#include <ctype.h>**

**int stack[100];**

**int top = -1;**

**void push(int item) {**

**if (top >= 99) {**

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

**exit(1);**

**}**

**stack[++top] = item;**

**}**

**int pop() {**

**if (top == -1) {**

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

**exit(1);**

**}**

**return stack[top--];**

**}**

**int evaluatePostfix(char \*postfix) {**

**int i, x1, x2;**

**i = 0;**

**while (postfix[i] != '\0') {**

**if (isdigit(postfix[i])) {**

**push(postfix[i] - '0'); // convert char to int**

**} else {**

**x2 = pop();**

**x1 = pop();**

**switch (postfix[i]) {**

**case '+':**

**push(x1 + x2);**

**break;**

**case '-':**

**push(x1 - x2);**

**break;**

**case '\*':**

**push(x1 \* x2);**

**break;**

**case '/':**

**if (x2 == 0) {**

**printf("Division by zero error\n");**

**exit(1);**

**}**

**push(x1 / x2);**

**break;**

**default:**

**printf("Invalid operator\n");**

**exit(1);**

**}**

**}**

**i++;**

**}**

**if (top != 0) {**

**printf("Invalid expression\n");**

**exit(1);**

**}**

**return stack[top];**

**}**

**int main() {**

**char postfix[100];**

**printf("Enter postfix expression: ");**

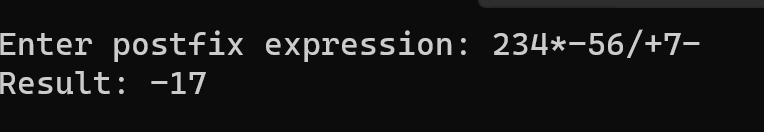
**gets(postfix);**

**int result = evaluatePostfix(postfix);**

**printf("Result: %d\n", result);**

**return 0;**

**}**

****

**4)**

#include <stdio.h>

void move(int n, int source, int destination, int auxiliary) {

while (n > 0) {

if (n % 2 == 0) {

printf(“Move disk %d from source %d to destination %d\n”, n, source, destination);

n--;

}

else {

printf(“Move disk %d from source %d to destination %d\n”, n, source, auxiliary);

printf(“Move disk %d from source %d to destination %d\n”, n – 1, source, destination);

printf(“Move disk %d from source %d to destination %d\n”, n, auxiliary, source);

n -= 2;

}

}

}

int main() {

int n;

printf(“Enter the number of disks: “);

scanf(“%d”, &n);

move(n, 1, 3, 2);

return 0;

}

