#include <stdio.h>

#include <stdbool.h>

typedef struct

{

char items[100];

int front;

int rear;

} Queue;

void initialize\_Queue(Queue \*q)

{

q->front = -1;

q->rear = -1;

}

bool isFull(Queue \*q)

{

return (q->rear == 99);

}

bool isEmpty(Queue \*q)

{

return (q->front == -1 && q->rear == -1);

}

void enqueue(Queue \*q, char value)

{

if (isFull(q))

{

printf("Queue is full\n");

return;

}

else if (isEmpty(q))

{

q->front = q->rear = 0;

}

else

{

q->rear++;

}

q->items[q->rear] = value;

}

char dequeue(Queue \*q)

{

char item;

if (isEmpty(q))

{

printf("Queue is empty\n");

return -1;

}

else if (q->front == q->rear)

{

item = q->items[q->front];

q->front = q->rear = -1;

}

else

{

item = q->items[q->front];

q->front++;

}

return item;

}

void print\_expression(char exp[])

{

Queue exps\_inside\_bracket, exps\_mult\_div, exps\_add\_sub;

initialize\_Queue(&exps\_inside\_bracket);

initialize\_Queue(&exps\_mult\_div);

initialize\_Queue(&exps\_add\_sub);

int i = 0;

bool bracket = false;

while (exp[i] != '\0')

{

if (exp[i] == '(')

{

bracket = true;

i++;

while (exp[i] != ')')

{

if (exp[i] == '\*' || exp[i] == '/')

{

printf("Do %s inside brackets()\n", (exp[i] == '\*') ? "multiplication" : "division");

}

else if (exp[i] == '+' || exp[i] == '-')

{

enqueue(&exps\_inside\_bracket, exp[i]);

}

i++;

}

while (!isEmpty(&exps\_inside\_bracket))

{

printf("Do %s inside brackets()\n", (dequeue(&exps\_inside\_bracket) == '+') ? "addition" : "subtraction");

}

}

bracket = false;

if (exp[i] == '\*' || exp[i] == '/')

{

enqueue(&exps\_mult\_div, exp[i]);

}

else if (exp[i] == '+' || exp[i] == '-')

{

enqueue(&exps\_add\_sub, exp[i]);

}

i++;

}

while (!isEmpty(&exps\_mult\_div))

{

printf("Do %s\n", (dequeue(&exps\_mult\_div) == '\*') ? "multiplication" : "division");

}

while (!isEmpty(&exps\_add\_sub))

{

printf("Do %s\n", (dequeue(&exps\_add\_sub) == '+') ? "addition" : "subtraction");

}

}

int main()

{

for (int i=0 ; i<3 ; i++)

{

char exp[100];

printf("Enter the Expression: ");

scanf("%s", exp);

print\_expression(exp);

printf("\n");

}

return 0;

}