```c
#include<stdio.h>
#include<stdlib.h>
#include<stdbool.h>
#include <ctype.h>

char stack[30],postfix[30],TAC[30][6],addr='Z';
int TOP=-1, p=0, t=0;

bool isEmpty()
{
	if (TOP==-1)
		return true;
	return false;
}
void push(char ch)
{
	stack[++TOP] = ch;
}
char pop()
{
	if(isEmpty()) {
		printf("Invalid expression\n");
		exit(0);
	}
	return stack[TOP--];
}
int pre(char ch)
{
	switch(ch) {
		case '+':
		case '-': return 1;
		case '*':
		case '/': return 2;
		default : return 0;
	}
}
void threeAddrCode()
{
	for (int i = 0; i < p; i += 1)
	{
		char ch = postfix[i];
		if (isalnum(ch))
			push(ch);
		else
		{
			TAC[t][4] = pop();
			TAC[t][3] = ch;
			TAC[t][2] = pop();
			TAC[t][1] = '=';
			TAC[t++][0] = addr;
			push(addr--);
		}
	}
}

int main()
{
	char expr[30];
	printf("Enter the expression: ");
	scanf("%s",expr);
	in_post(expr);
	threeAddrCode();
	printf("Intermediate Code:\n");
	for (int i = 0; i < t; i += 1)
	{
		printf("%s\n",TAC[i]);
	}
}
```

```
void in_post(char expr[30])
{
	int i=0, op;
	char ch;
	for (i=0; expr[i]!='\0'; i++)
	{
		ch = expr[i];
		if (isalnum(ch))
		{
			postfix[p++] = ch;
		}
		else if(ch=='(')
		{
			push('(');
		}
		else if(ch==')')
		{
			op = pop();
			while(op!='(')
			{
				postfix[p++] = op;
				op = pop();
			}
		}
		else
		{
			while (!isEmpty())
			{
				op = pop();
				if (pre(ch)<=pre(op))
				{
					postfix[p++] = op;
				}
				else
				{
					push(op);
					break;
				}
			}
			push(ch);
		}
	}
	while(!isEmpty())
	{
		postfix[p++] = pop();
	}
}
```