```c
#include <stdio.h>
#include <stdlib.h>
typedef struct s_op
{
	int status;
	int ans;
}				t_op;

int stack[256];
int pointer = -1;

int pop()
{
	return stack[pointer--];
}

void push(int num)
{
	stack[++pointer] = num;
}

int isemp()
{
	return pointer == -1;
}

int isspc(char c)
{
	return c == ' ' || c == '\t' || c == '\r'
		|| c == '\f' || c == '\n' || c == '\v';
}

int isdig(char c)
{
	return (c >= '0' && c <= '9');
}

int isop(char c)
{
	return c == '*' || c == '/' || c == '+'
		|| c == '-' || c == '%';
}
```

```c
t_op *doop(char op)
{
	t_op *res = malloc(sizeof(t_op));
	res->status = 1;
	int num1;
	int num2;
	if (!isemp())
		num1 = pop();
	else
	{
		res->status = 0;
		return (res);
	}
	if (!isemp())
		num2 = pop();
	else
	{
		res->status = 0;
		return (res);
	}

	if (op == '+')
		res->ans = num1 + num2;
	else if (op == '-')
		res->ans = num2 - num1;
	else if (op == '*')
		res->ans = num1 * num2;
	else if (op == '/')
	{
		if (num1 == 0)
			res->status = 0;
		else
			res->ans = num2 / num1;
	}
	else
	{
		if (num1 == 0)
			res->status = 0;
		else
			res->ans = num2 % num1;
	}
	return res;
}
```

```c
int calc(char *equ)
{
	int i = 0;
	t_op *res;
	while(equ[i])
	{
		while (isspc(equ[i]))
			i++;
		if (isop(equ[i]) && (!equ[i + 1] || isspc(equ[i + 1])))
		{
			res = doop(equ[i]);
			if (res->status == 0)
				return 0;
			else
				push(res->ans);
		}
		while (isspc(equ[i]))
			i++;
		if (isdig(equ[i]) || (equ[i] == '-' && isdig(equ[i + 1])))
		{
			push(atoi(equ + i));
			if (equ[i] == '-')
				i++;
		}
		while (isdig(equ[i]))
			i++;
		i++;
	}
	int ans = pop();
	if (isemp())
		printf("%d\n", ans);
	else
		return (0);
	return (1);
}

int main(int ac, char **av)
{
	if (ac != 2)
	{
		printf("Error\n");
		return (0);
	}
	else
	{
		if (!calc(av[1]))
			printf("Error\n");
	}
	return (0);
}
```