UNIVERSIDAD AUTÓNOMA DE CHIHUAHUA
Facultad de Ingeniería

Ingeniería en Ciencias de la Computación

TEORÍA DE LA COMPUTACIÓN
Programación de un intérprete parte 1

Trabajo de: ADRIAN A. GONZÁLEZ DOMÍNGUEZ *[359834]*
Asesor: MARIO ANDRES CUEVAS GUTIERREZ

1 de octubre de 2024

Teoría de la computación | Septiembre 29, 2024

Programación de un interprete

Adrian A. González Dominguez 359834

```
#include <stdio.h>
#include <ctype.h>
#include "stack.h"
FILE *input_file;
snode_t *stack=NULL;
int S(), E(), M(), F(), N();
int main(int argc, char **args){
  if(argc==2) input_file=fopen(args[1], "r");
  else input_file=stdin;
  while(!feof(input_file)){
    if(S()){
      int *result=pop(&stack);
      printf("%d\n", *result);
      printf("OK");
    } else while(
      !feof(input_file)&&(getc(input_file))!='\n'
    );
    putc('\n', stdout);
  }
}
```

```
int S(){
  if(E()){
    char c=getc(input_file);
    if(c=='\n') return 1;
  }
  return 0;
}
int E(){
  if(M()){
    char c=getc(input_file);
    if(c=='+'){
      if(E()){
        int *b=pop(&stack);
        int *a=pop(&stack);
        int *c=malloc(sizeof(int));
        *c=*a+*b;
        push(stack, c);
        free(a);
        free(b);
        return 1;
      }
      fprintf(stderr, "Se esperaba algo después de \'+\'");
      return 0;
    }
```

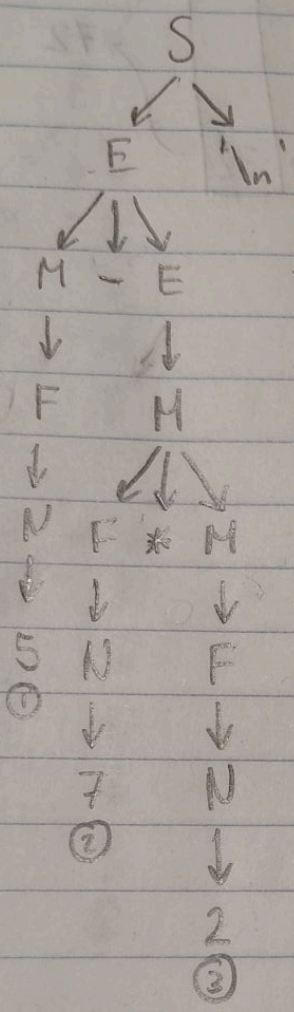
```
    if(c=='-'){
      if(E()){
        int *b=pop(&stack);
        int *a=pop(&stack);
        int *c=malloc(sizeof(int));
        *c=*a-*b;
        push(&stack, c);
        free(a);
        free(b);
        return 1;
      }
      fprintf(stderr, "Se esperaba algo despues de \'-\'");
      return 0;
    }
    ungetc(c, input_file);
    return 1;
  }
  return 0;
}
int M(){
  if(F()){
    char c=getc(input_file);
    if(c=='*'){
      if(M()){
        int *b=pop(&stack);
        int *a=pop(&stack);
        int *c=malloc(sizeof(int));
        *c=*a**b;
```

```
        push(&stack, c);
        free(a);
        free(b);
        return 1;
      }
      fprintf(stderr, "Se esperaba algo después de \'*\'");
      return 0;
    }
    if(c=='/'){
      if(M()){
        int *b = pop(&stack);
        int *a = pop(&stack);
        int *c = malloc(sizeof(int));
        *c = *a / *b;
        push(&stack, c);
        free(a);
        free(b);
        return 1;
      }
      fprintf(stderr, "Se esperaba algo después de \'/\'");
      return 0;
    }
    ungetc(c, input_file);
    return 1;
  }
  return 0;
}
```

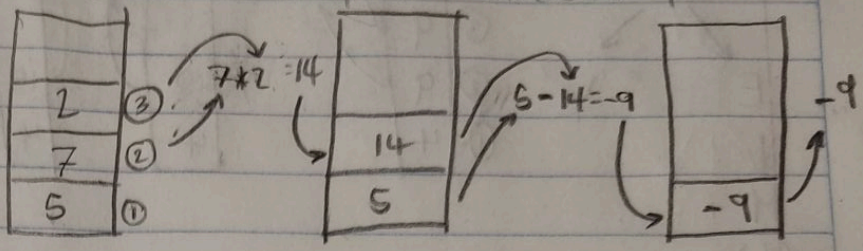
```
int F(){
  if(NC()) return 1;
  char c=getc(input_file);
  if(c=='-'){
    if(F()){
      int *value=top(&stack);
      *value*=-1;
      return 1;
    }
    fprintf(stderr, "Se esperaba un factor después de -");
    return 0;
  }
  if(c=='('){
    if(E()){
      char c=getc(input_file);
      if(c==')'){
        return 1;
      }
      fprintf(stderr, "Se esperaba un ')'");
      ungetc(c, input_file);
      return 0;
    }
  }
  ungetc(c, input_file);
  return 0;
}
```

```
int N(){
  char c=getc(inpat_file);
  if(isdigit(c)){
   int *value=malloc(sizeof(int));
   *value=c-48;
   push(&stack, value);
   return 1;
  }
  ungetc(c, inpat_file);
  return 0;
}
```

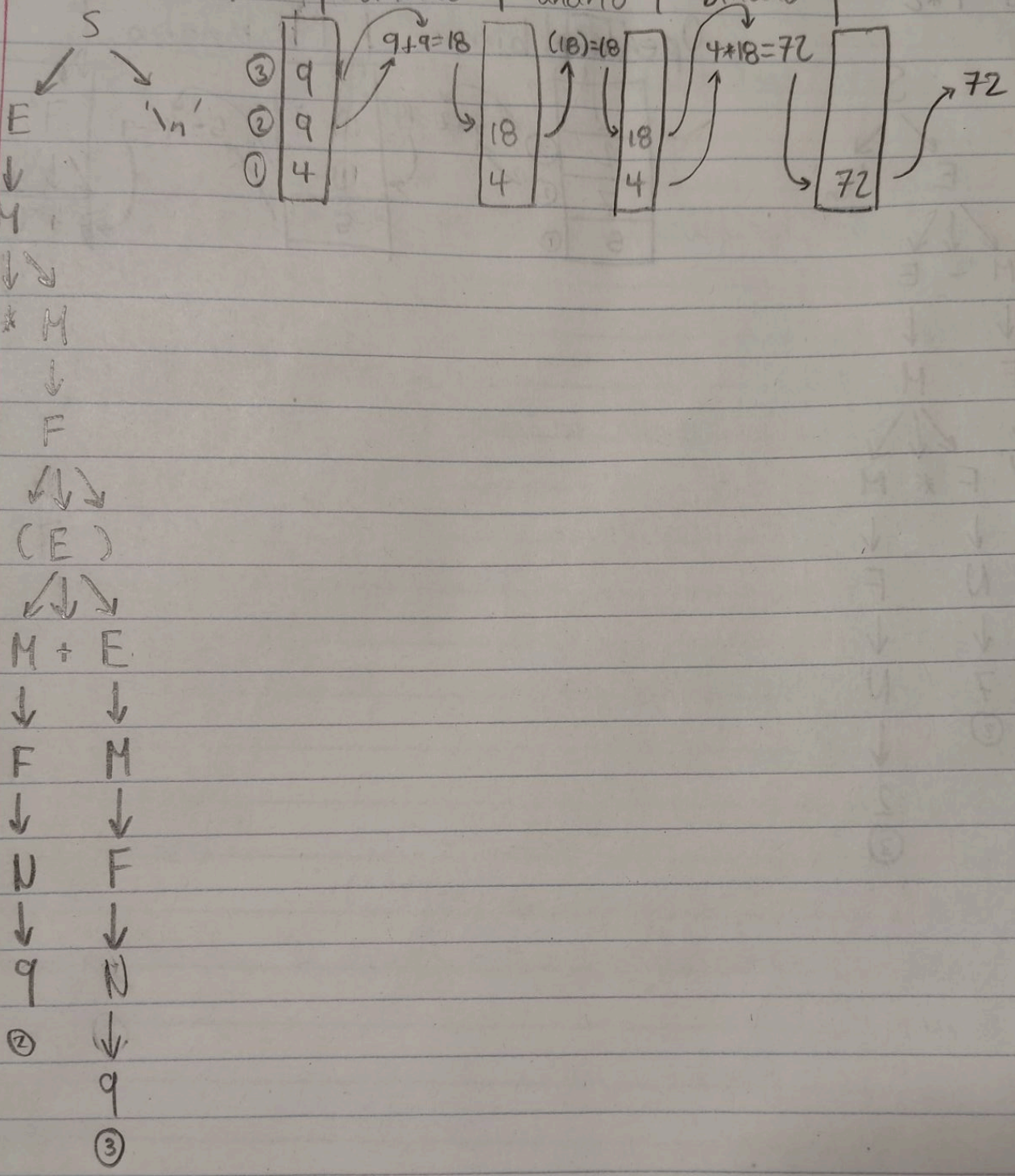
5-7*2



Operador binario | binario



4*(9+9)


S
E
'\n'
M
F
* M
F
N
4
(E)
M + E
F
M
N
F
9
N
9
Operador
binario
unario
binario
③ 9
② 9
① 4
9+9=18
18
4
(18)=18
18
4
4*18=72
72
72

-5--3

S

E '\n'

M - E

F M

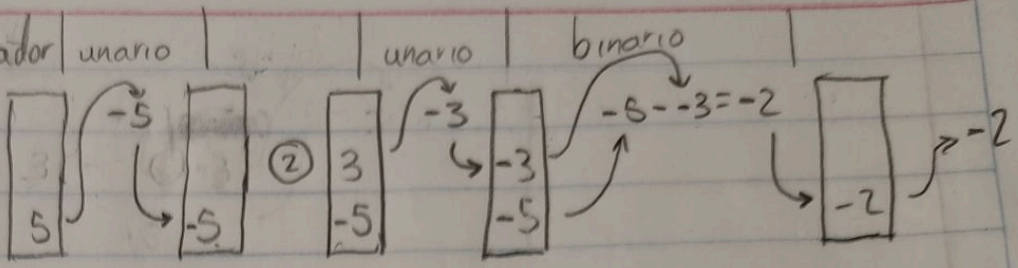
F F

F F

N - F

E N

① 3



Pruebas realizadas

```
$ gcc stack.c TOC.September_24_2024.c -o TOC.September_24_2024 && ./TOC.September_24_2024
5-7*2
-9
OK
4*(9+9)
72
OK
-5--3
-2
-9
OK
4*(9+9)
72
OK
-5--3
-2
4*(9+9)
72
OK
-5--3
-2
-5--3
-2
OK
```