

# Relatório T2

Érica Oliveira Regnier 2211893

Hanna Epelboim Assunção 2310289

3WB

## ★ Main 1:

```
1 #include <stdlib.h>
2 #include "compilalinb.h"
3
4 #define TAM_CODIGO 2000
5
6 int main(void) {
7     unsigned char codigo[TAM_CODIGO];
8     FILE* myfp;
9     funcp f;
10    //char nome_arquivo = argv[1];
11    int retorno;
12
13    if ((myfp = fopen ("teste.txt", "r")) == NULL) {
14        perror ("nao conseguiu abrir arquivo!");
15        exit(1);
16    }
17
18    f = compilaLinB(myfp, codigo);
19
20    retorno = (f)(10,20); /* se a função não recebe parametros, são descartados */
21
22    printf("resultado = %d\n",retorno);
23
24 }
```

Primeiro teste:

Saída esperada: 11

```
1 v1 <= p1 + $1
2 ret
3 |
```

```
eriquita@batatatraste:~/Documents/T2$ gcc -Wall -Wa,--execstack -o t2 maint2.c  
compilalinb.c  
eriquita@batatatraste:~/Documents/T2$ ./t2  
resultado = 11  
eriquita@batatatraste:~/Documents/T2$
```

Segundo teste:

Saída esperada: 3628800

10 !=  
3628800

```
1 v1 <= $0 + $1  
2 if p1 4  
3 ret  
4 v1 <= v1 * p1  
5 p1 <= p1 - $1  
6 if p1 4  
7 ret
```

```
resultado = 11  
eriquita@batatatraste:~/Documents/T2$ gcc -Wall -Wa,--execstack -o t2 maint2.c compilalinb.c  
eriquita@batatatraste:~/Documents/T2$ ./t2  
resultado = 3628800  
eriquita@batatatraste:~/Documents/T2$ gcc -Wall -Wa,--execstack -o t2 maint2.c compilalinb.c
```

Terceiro teste:

Saída esperada: -300

```
1 v1 <= p1 + p2
2 p1 <= p1 - p2
3 v1 <= v1 * p1
4 ret
5
6 |
```

```
eriquita@batatatraste:~/Documents/T2$ gcc -Wall -Wa,--execstack -o t2 maint2.c
compilalinb.c
eriquita@batatatraste:~/Documents/T2$ ./t2
resultado = -300
eriquita@batatatraste:~/Documents/T2$
```

## ★ Main 2:

```
1 #include <stdlib.h>
2 #include "compilalinb.h"
3
4 #define TAM_CODIGO 2000
5
6 int main(void) {
7     unsigned char codigo[TAM_CODIGO];
8     FILE* myfp;
9     funcp f;
10    //char nome_arquivo = argv[1];
11    int retorno;
12
13    if ((myfp = fopen ("teste.txt", "r")) == NULL) {
14        perror ("nao conseguiu abrir arquivo!");
15        exit(1);
16    }
17
18    f = compilaLinB(myfp, codigo);
19
20    retorno = (f)(3,1); /* se a função não recebe parametros, são descartados */
21
22    printf("resultado = %d\n",retorno);
23
24 }
25
26
```

Primeiro teste:

Saida esperada: 14

```
1 v1 <= $0 + $0
2 v2 <= $1 + $0
3 p1 <= p1 + $1
4 v3 <= p1 - v2
5 if v3 7
6 ret
7 v3 <= v2 * v2
8 v1 <= v1 + v3
9 v2 <= v2 + $1
0 if v2 4
1
```

```
eriquita@batatatriste:~/Documents/T2$ gcc -Wall -Wa,--execstack -o t2 maint2.c  
compilalib.c  
eriquita@batatatriste:~/Documents/T2$ ./t2  
resultado = 14  
eriquita@batatatriste:~/Documents/T2$
```

Segundo teste:

Saida esperada: 6

```
1 v1 <= $0 + $1  
2 if p1 4  
3 ret  
4 v1 <= v1 * p1  
5 p1 <= p1 - $1  
6 if p1 4  
7 ret
```

```
eriquita@batatatriste:~/Documents/T2$ gcc -Wall -Wa,--execstack -o t2 maint2.c  
compilalib.c  
eriquita@batatatriste:~/Documents/T2$ ./t2  
resultado = 6  
eriquita@batatatriste:~/Documents/T2$
```

Terceiro teste:

Saida esperada: 4

```
1 v1 <= p1 + $1  
2 ret  
3
```

```

eriquita@batatatrister:~/Documents/T2$ gcc -Wall -Wextra --execstack -o t2 maint2.c
compilalinb.c
eriquita@batatatrister:~/Documents/T2$ ./t2
resultado = 4
eriquita@batatatrister:~/Documents/T2$

```

### ★ Main 3:

```

1 #include <stdlib.h>
2 #include "compilalinb.h"
3
4 #define TAM_CODIGO 2000
5
6 int main(void) {
7     unsigned char codigo[TAM_CODIGO];
8     FILE* myfp;
9     funcp f;
10    //char nome_arquivo = argv[1];
11    int retorno;
12
13    if ((myfp = fopen ("teste.txt", "r")) == NULL) {
14        perror ("nao conseguiu abrir arquivo!");
15        exit(1);
16    }
17
18    f = compilaLinB(myfp, codigo);
19
20    retorno = (f)(1000,2000); /* se a função não recebe parametros, são descartados */
21
22    printf("resultado = %d\n",retorno);
23
24 }
25

```

Primeiro teste:

Saída esperada: 3000

```
1 v1 <= p1 + p2
2 ret
3
```

```
eriquita@batatatraste:~/Documents/T2$ gcc -Wall -Wa,--execstack -o t2 maint2.c
compilalib.c
eriquita@batatatraste:~/Documents/T2$ ./t2
resultado = 3000
eriquita@batatatraste:~/Documents/T2$
```

Segundo teste:

Saída esperada: 2000000

```
1 v1 <= p1 * p2
2 ret
3
```

```
compilalib.c
eriquita@batatatraste:~/Documents/T2$ ./t2
resultado = 2000000
eriquita@batatatraste:~/Documents/T2$
```

Terceiro teste:

Saída esperada: -1000

```
1 v1 <= p1 - p2
2 ret
3
```

```
resultado = 2000000
eriquita@batatatraste:~/Documents/T2$ gcc -Wall -Wa,--execstack -o t2 maint2.c
compilalinb.c
eriquita@batatatraste:~/Documents/T2$ ./t2
resultado = -1000
eriquita@batatatraste:~/Documents/T2$
```

Quarto teste:

Saída esperada: 1000

```
1 v1 <= p2 - p1
2 ret
3
```

```
resultado = -1000
eriquita@batatatraste:~/Documents/T2$ gcc -Wall -Wa,--execstack -o t2 maint2.c
compilalinb.c
eriquita@batatatraste:~/Documents/T2$ ./t2
resultado = 1000
```



### ★ Main 4:

```
teste.txt  x  compilalib.h  x  compilalib.c
1 #include <stdlib.h>
2 #include "compilalib.h"
3
4 #define TAM_CODIGO 2000
5
6 int main(void) {
7     unsigned char codigo[TAM_CODIGO];
8     FILE* myfp;
9     funcp f;
10    //char nome_arquivo = argv[1];
11    int retorno;
12
13    if ((myfp = fopen ("teste.txt", "r")) == NULL) {
14        perror ("nao conseguiu abrir arquivo!");
15        exit(1);
16    }
17
18    f = compilaLinB(myfp, codigo);
19
20    retorno = (f)(1,0); /* se a função não recebe parametros, são descartados */
21
22    printf("resultado = %d\n",retorno);
23
24 }
25
```

Primeiro teste:

Saída esperada: 11

```
1 v1 <= p1 + $1
2 v2 <= p2 * $5
3 v3 <= p2 - $-9
4 v4 <= v1 + v3
5 v1 <= v4 + v2
6 ret
7 |
```

Segundo teste:

### ★ Main 5:

```
teste.txt  ×  compilalinb.h  ×  compilalinb.c
1 #include <stdlib.h>
2 #include "compilalinb.h"
3
4 #define TAM_CODIGO 2000
5
6 int main(void) {
7     unsigned char codigo[TAM_CODIGO];
8     FILE* myfp;
9     funcp f;
10    //char nome_arquivo = argv[1];
11    int retorno;
12
13    if ((myfp = fopen ("teste.txt", "r")) == NULL) {
14        perror ("nao conseguiu abrir arquivo!");
15        exit(1);
16    }
17
18    f = compilaLinB(myfp, codigo);
19
20    retorno = (f)(2,3); /* se a função não recebe parametros, são descartados */
21
22    printf("resultado = %d\n",retorno);
23
24 }
```

Primeiro teste:

Saída esperada: 3

```
1 v1 <= $1 + $0
2 if p2 4
3 ret
4 v1 <= v1 + p1
5 p2 <= p2 - $1
6 ret
7 |
```

```
eriquita@batatatraste:~/Documents/T2$ gcc -Wall -Wa,--execstack -o t2 maint2.c  
compilalinb.c  
eriquita@batatatraste:~/Documents/T2$ ./t2  
resultado = 3  
eriquita@batatatraste:~/Documents/T2$
```

Segundo teste:

Saída esperada: 7

```
1 v1 <= $1 + $0  
2 if p2 4  
3 ret  
4 v1 <= v1 + p1  
5 p2 <= p2 - $1  
6 if p2 4  
7 ret  
8
```

```
eriquita@batatatraste:~/Documents/T2$ gcc -Wall -Wa,--execstack -o t2 maint2.c  
compilalinb.c  
eriquita@batatatraste:~/Documents/T2$ ./t2  
resultado = 7  
eriquita@batatatraste:~/Documents/T2$
```

### ★ Main 6:

```
1 #include <stdlib.h>
2 #include "compilalinb.h"
3
4 #define TAM_CODIGO 2000
5
6 int main(int argc, char *argv[]) {
7     unsigned char codigo[TAM_CODIGO];
8     FILE* myfp;
9     funcp f;
10    char *nome_arquivo = argv[1];
11    int retorno;
12
13    if ((myfp = fopen (nome_arquivo, "r")) == NULL) {
14        perror ("nao conseguiu abrir arquivo!");
15        exit(1);
16    }
17
18    f = compilaLinB(myfp, codigo);
19
20    retorno = (*f)(10,20); /* se a função não recebe parametros, são descartados */
21
22    printf("resultado = %d\n",retorno);
23
24 }
```

Primeiro teste:

Saída esperada: 1

```
1 v1 <= $1 + $0
2 ret
3 |
```

```
eriquita@batatatraste:~/Documents/T2$ gcc -Wall -Wa,--execstack -o t2 main.c compilalinb.c
eriquita@batatatraste:~/Documents/T2$ ./t2 cod.linb
resultado = 1
eriquita@batatatraste:~/Documents/T2$
```

Segundo teste:

Saída esperada: 3628800

```
1 v1 <= $0 + $1
2 if p1 4
3 ret
4 v1 <= v1 * p1
5 p1 <= p1 - $1
6 if p1 4
7 ret
8 |
```

```
resultado = 1
eriquita@batatatraste:~/Documents/T2$ gcc -Wall -Wa,--execstack -o t2 main.c compilalib.c
eriquita@batatatraste:~/Documents/T2$ ./t2 cod.lib
resultado = 3628800
eriquita@batatatraste:~/Documents/T2$
```

Observações: Quando não há v1, nosso compilador retorna um lixo.