Compilers Assignment 6

George Gkikas-Panousis (1820253069)
Anastasios Dyiolis (1820253075)

3-Address Code (TAC)

Below is the 3-address code (TAC) version of the provided `test.c` file. Each instruction corresponds to a line in the source file, annotated and simplified using the corresponding LLVM IR representation. Line numbers correspond to the source C code.

```
01: int a[10][20]; // global: no TAC needed
02: int b = 1; // global: b = 1
03: int f1(int x, int y) {
04: t1 = x + y // z = x + y
05: return t1
06: }

07: void f2() {
08: param "in f2\n"
      call Mars_PrintStr
09: return
10: }

11: int main() {
12: a1 = 1
13: a2 = 2
14: // res declared

15: t2 = (a1 == 0)
      res = t2

16: res = bitwise_not a1

17: t3 = a1 + a2
      res = t3

18: t4 = a1 % a2
      res = t4

19: t5 = a1 << a2
      res = t5

20: res = a1
      a1 = a1 + 1

21: a1 = a1 + 1
      res = a1

22: t6 = (a1 != 0)
      t7 = (a2 != 0)
      t8 = t6 && t7
      if t8 goto L1
      goto L2

23: L1: param a1
      param a2
```

```
      t9 = call f1
      res = t9
      goto LendIf

24: L2: t10 = (a1 == 0)
      if t10 goto L3
      goto L4

25: L3: param b
      param a2
      t11 = call f1
      res = t11
      goto LendIf

27: L4: call f2()

30: i = 0
Lloop: t12 = i < a1
      if not t12 goto LendFor

31: goto LendFor
32: goto LnextIter
33: res = res + 1

LnextIter:
      i = i + 1
      goto Lloop
LendFor:

35: Lk: goto Lk

37: return 0
38: }
```