Project Compilers

Ranime Hossam	1170039
Aya Magdy Essam	1170018
Mariam Amr	1170074
Aya Mahmoud Abdelfattah	1170049
Areej Khaled	1170233

Quadruples:

- **int x = 0**

  ```
  LD t0,0
  ST x,t0
  ```

- **y = x + z**

  ```
  LD t0,x
  LD t1,z
  ADD t0,t1
  ST y,t0
  ```

- **int sum(int x, int y){ return x + y }**
  void main(){ int a = sum(10, 20) a = sum(a, a)}

  ```
  pop t0
  ST y,t0
  pop t0
  ST x,t0
  LD t0,x
  LD t1,y
  Add t2,t0,t1
  push t2
  JMP d0
  L001:
  LD t3,10
  push t3
  LD t4,20
  push t4
  LD t5 ,L002
  push t5
  JMP L000
  L002:
  pop t5
  ST a,t5
  LD t0,a
  push t0
  LD t1,a
  push t1
  LD t2 ,L003
  push t2
  JMP L000
  L003:
  pop t2
  ST a,t2
  pop t0
  JMP t0
  ```

Note that: The last pop and jump is because we consider the main as a function, that is why we pop from the stack the address from which the function was called and jump to it.

- **if(x==0){print x}else{print y}**

```
LD t0, x
LD t1,0
cmpEQ t0,t1
jnz L000
LD t2,x
print t2
L001:
LD t3,y
print t3
L000:
```

- **enum day_of_the_week{ Mon,Sun}**

```
LD t0,0
ST Mon,t0
LD t0,1
ST Sun,t0
```

Test cases

Test case 1: Expressions:

```
void main()
{
  int x = 0
  int y = 10
  int z = x / y - x
  bool a = true
  bool b = false
  bool c = a && b
}

L000:
LD t0,0
ST x,t0
LD t0,10
ST y,t0
LD t0,x
LD t1,y
DIV t2,t0,t1
LD t3,x
```

```
sub t4,t2,t3
ST z,t4
ST a,
ST b,
LD t0,a
LD t1,b
AND t0,t1
ST c,t2
pop t0
JMP t0
```

Test case 2: enum_error

```
enum day_of_the_week
{
  Mon,
  Sun
};

void main()
{
  int x = Tue
}
```

```
the variable is undeclared
Type mismatch
```

Test case 3: enum

```
enum day_of_the_week
{
  Mon,
  Sun
};

void main()
{
  int x = Mon
}
```

```
LD t0,0
ST Mon,t0
LD t0,1
ST Sun,t0
L000:
LD t0,Mon
ST x,t0
pop t0
JMP t0
```

Test case 4: If

```
void main()
{
  int x = 10
  int b
  if (x == 10)
  {
    b = 10
  }
  else
  {
   if (x == 5){
    b = 5
    }
   else
    {
    b = x
     }
  }
}
```

```
L000:
LD t0,10
ST x,t0
LD t0,x
LD t1,10
compEQ t0,t1
jnz     L001
LD t0,10
ST b,t0
jmp     L002
L001:
LD t0,x
LD t1,5
compEQ t0,t1
jnz     L003
LD t0,5
ST b,t0
jmp     L004
L003:
LD t0,x
ST b,t0
L004:
L004:
pop t0
JMP t0
```

Test Case 5: Semantic errors

```
void main()
{
  int x=0
  string a="dd"
  x = a
}
L000:
LD t0,0
ST x,t0
ST a,
LD t0,a
pop t1
JMP t1
```

Test Case 6: Syntax error

```
void main()
{
  int 2s = 10
}
syntax error in line 3
```

Test Case 7: Loops

```
void main()
{

  for (int i = 0; i < 10; i++)
  {
    int b = 10
  }

  int x = 0

  while (x < 20)
  {
    x++
  }

  x = 0

  repeat
  {
    x++
  } until(x >= 20)
  switch (x)
  {
  case 1:
    x = 10 break

  default:
    break
  }
}

L000:
LD t0,0
ST i,t0
LD t0,i
LD t1,10
compLT t0,t1
LD t0,10
ST b,t0
LD t0,0
ST x,t0
L001:
LD t0,x
LD t1,20
compLT t0,t1
jnz       L002
jmp       L001
L002:
LD t0,0
ST x,t0
L003:
LD t0,x
LD t1,20
compGE t0,t1
jnz       L004
jmp       L003
L004:
LD t0,x
compEQ t0,1
jnz       L006
LD t0,10
ST x,t0
jmp       L005
L006:
L005:
pop t0
JMP t0
```

Test Case 8: Functions

```
int sum(int x, int y)
{
  return x + y
}

void main()
{
  int a = sum(10, 20)
  a = sum(a, a)
}
L000:
pop d0
pop t0
ST y,t0
pop t0
ST x,t0
LD t0,x
LD t1,y
Add t2,t0,t1
push t2
JMP d0
L001:
LD t3,10
push t3
LD t4,20
push t4
LD t5 ,L002
push t5
JMP L000
L002:
pop t5
ST a,t5
LD t0,a
push t0
LD t1,a
push t1
LD t2 ,L003
push t2
JMP L000
L003:
pop t2
ST a,t2
pop t0
JMP t0
```