Ishita Hardasmalani

C14-2103058

EXPERIMENT NO.9

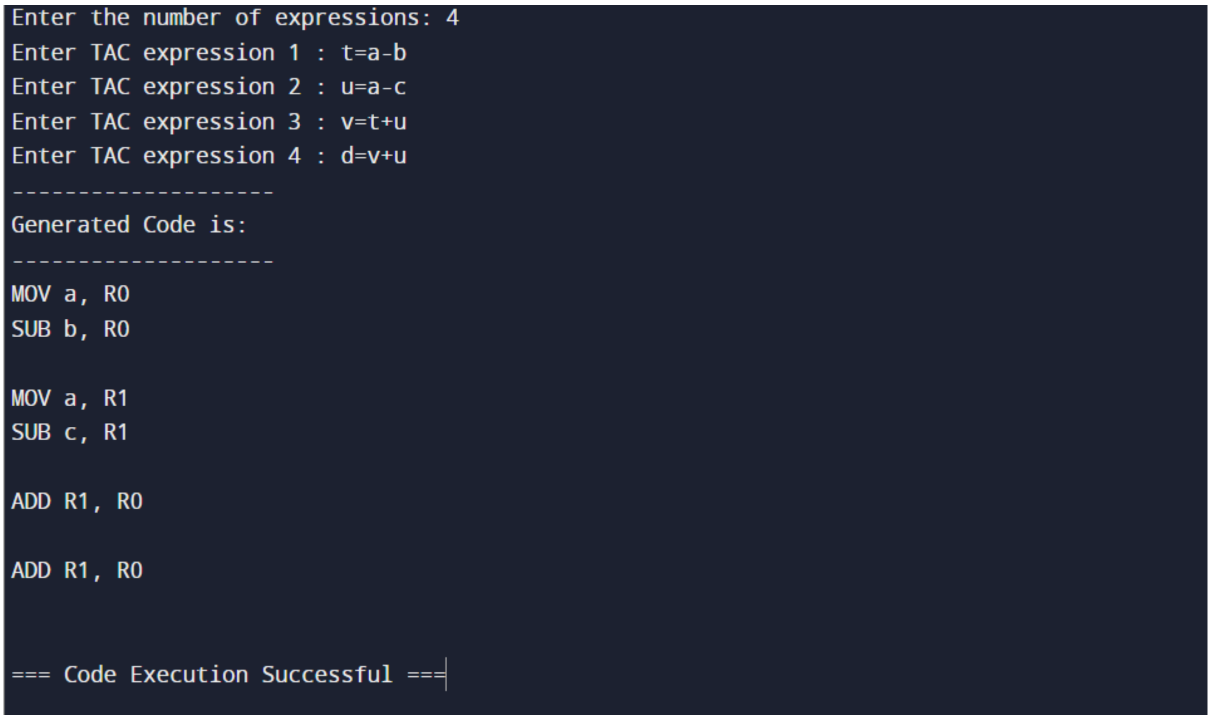
Aim : Write a program to implement code generation

Code:

```
def main():
  n = int(input("Enter the number of expressions: "))
   expressions = {}
  expr_registers = {}
  codeGen = []
  count = 0
 opcodes = {
     "+": "ADD",
     "-": "SUB",
     "*": "MUL",
     "/": "DIV"
  }

  for i in range(n):
     exp = input(f"Enter TAC expression {i + 1} : ")

     lhs, rhs = exp.split('=')
     expressions[lhs] = rhs

     op1 = rhs[0]
     op = rhs[1]
     op2 = rhs[2]
```

```
        expr_registers[lhs] = f"R{count}"

        if expr_registers.get(op1) is None or expr_registers.get(op2) is None:
            code = f"MOV {op1}, R{count} \n{opcodes[op]} {op2}, R{count}"
            codeGen.append(code)
            count += 1
        else:
            expr_registers[lhs] = expr_registers[op1]
            code = f"{opcodes[op]} {expr_registers[op2]}, {expr_registers[op1]}"
            codeGen.append(code)
    # print(expr_registers)

    print("--"*10)
    print("Generated Code is: ")
    print("--" * 10)
    for code in codeGen:
        print(code)
        print()
if __name__ == "__main__":
    main()
```

Output:



```
Enter the number of expressions: 4
Enter TAC expression 1 : t=a-b
Enter TAC expression 2 : u=a-c
Enter TAC expression 3 : v=t+u
Enter TAC expression 4 : d=v+u
--------------------
Generated Code is:
--------------------
MOV a, R0
SUB b, R0

MOV a, R1
SUB c, R1

ADD R1, R0

ADD R1, R0


=== Code Execution Successful ===
```