Compiler Construction

Assignment – 12

Q) Generation of target code for given 3-address code.

Code:

```
class TargetCodeGenerator:
    def __init__(self):
        self.target_code = []

    def generate_target_code(self, three_address_code):
        for instruction in three_address_code:
            op, arg1, arg2, result = instruction
            target_instruction = self.map_to_target(op, arg1, arg2, result)
            self.target_code.append(target_instruction)

    def map_to_target(self, op, arg1, arg2, result):
        # Basic mapping, assuming a simple architecture
        if op == "+":
            return f"ADD {result}, {arg1}, {arg2}"
        elif op == "-":
            return f"SUB {result}, {arg1}, {arg2}"
        elif op == "*":
            return f"MUL {result}, {arg1}, {arg2}"
        elif op == "/":
            return f"DIV {result}, {arg1}, {arg2}"
        else:
            raise ValueError(f"Unsupported operation: {op}")
```

```
    def get_target_code(self):
        return self.target_code


# Take user input for 3-address code
three_address_code = []
while True:
    input_str = input("Enter 3-address code instruction (or 'done' to finish): ").strip()
    if input_str.lower() == 'done':
        break

    # Split the input into components
    instruction_parts = input_str.split()
    if len(instruction_parts) == 4:
        three_address_code.append(tuple(instruction_parts))
    else:
        print("Invalid input. Please enter a valid 3-address code instruction.")

# Create a TargetCodeGenerator instance
generator = TargetCodeGenerator()

# Generate target code
generator.generate_target_code(three_address_code)

# Get the generated target code
target_code = generator.get_target_code()

# Print the generated target code
print("\nGenerated Target Code:")
for instruction in target_code:
```

```
print(instruction)
```

OUTPUT:

```
========
Enter 3-address code instruction (or 'done' to finish): + a b temp1
Enter 3-address code instruction (or 'done' to finish): * temp1 c result
Enter 3-address code instruction (or 'done' to finish): / result d final
_result
Enter 3-address code instruction (or 'done' to finish): done

Generated Target Code:
ADD temp1, a, b
MUL result, temp1, c
DIV final_result, result, d
```