Lab - 07

ACT : Advanced Compiler Techniques

AIM : Perform local copy propagation

Name : Ambalia Harshit
Roll no : MT001
Date : 13 Oct 2023

Question 01 : Given 3 address codes, Perform local copy propagation.

Input : Simple 3 address code
Output : Operations

Code :

```
## LAB - 07

# NAME : AMBALIA HARSHIT
# SUBJECT : ACT
# ROLL NO. : MT001
# AIM : Local copy Propogation

# Input : Number of statements, basic expressions.
# Output : statements after local copy propagation

# Restrictions :
# Can perform LCP only on following types of the statements:
#   variable = value/variable
#   variable = value/variable [+, -, /, *, %] value/variable

def local_copy_propagation(statements):
   my_dict = {}
```

```
    answer = []
    for _,stmt in enumerate(statements):
        if(len(stmt)==3):
            # Check if RHS is in dictionary if not add it. if yes update
it.
            if(stmt[2] in my_dict):
                answer.append(stmt[0]+"="+my_dict[stmt[2]])
                my_dict[stmt[0]] = my_dict[stmt[2]]
            else:
                answer.append(stmt)
                my_dict[stmt[0]] = stmt[2]
                if stmt[0] in my_dict.values():
                    key_to_remove = None
                    for key, value in my_dict.items():
                        if value == stmt[0]:
                            key_to_remove = key
                            break
                    if key_to_remove is not None:
                        my_dict.pop(key_to_remove)

        if(len(stmt)==5):
            if(stmt[2] in my_dict and stmt[4] in my_dict):

answer.append(stmt[0]+"="+my_dict[stmt[2]]+stmt[3]+my_dict[stmt[4]])
            else:
                if(stmt[2] in my_dict):

answer.append(stmt[0]+"="+my_dict[stmt[2]]+stmt[3]+stmt[4])
                if(stmt[4] in my_dict):

answer.append(stmt[0]+"="+stmt[2]+stmt[3]+my_dict[stmt[4]])
            if(stmt[0] in my_dict):
                my_dict.pop(stmt[0])
        print(stmt, my_dict, answer)

def main():
    no_of_stat = int(input("Enter number of statements : "))
    statements = []
    for _ in range(no_of_stat):
        statements.append(input("Enter Expression : "))
```

```
    local_copy_propagation(statements)

if __name__=="__main__":
    main()
```

Restrictions :

- Can use only following type statements as input statements.
    - Variable = value/variable
    - Variable = value/variable [+, -, /, *,] value/variable

Output Screenshot :