11/04/23

EXPERIMENT-12

DESIGN AND DEVELOP A SIMPLE CODE GENERATOR

AIM: To design and develop a simple code generation. [back end of compiler]

ALGORITHM:

1. Start
2. Get addresses Code Sequence
3. Determine current location of 3 using address
4. If current location is not ready, generate MOV (R0)
5. Update address of A (for 2nd Operand)
6. If current value of B and C is null, Exit
7. If they generate operator C, A, B, ADDR
8. Stop & move instruction in memory.
9. Stop.

PROGRAM:

```
#include <stdio.h>
#include <string.h>
void main()
{
  char icode[10][30], str[20], opr[10];
  int i=0;
  Printf("\n Enter set of Intermediate Code : \n");
  do { scanf("%s", icode[i]);
    } while (strcmp(icode[i++], "exit")!=0)
printf("\n Target Code generation");
printf("\n ********");
  i=0;
  do {
  strcpy(str, icode[i]);
    switch (str[3]) {
```

Output:

Enter the set of intermediate code (terminated by exit):

4+6

10

exit

Target code generation

```
**********

    MOV 6,R0
;,R0
    MOV R0,4
    MOV 0,R1
;,R1
    MOV R1,0
```

O/p verified

11/4/23.

```
{   case '+':
    strcpy(opr, "ADD");
    break;
  Case '-':
  strcpy (opr "SUB");
  break;
  case '*';
  Strcpy (opr "MUL");
  break;
  Case '/':
    Strcpy (opr "DIV");
    break }
Printf ("\n\t Mov%c, R%d", str[2], i);
Printf ("\n %s%c ;R%d", opr, str[1], i);
Printf ("\n\t Mov R%d, %c", i, str[0]); }
where (strcmp (icode [++P], "exit") != 0);
}
```

RESULT: Simple code generation is implimented using above code.