Ex No: 9

Date:

IMPLEMENT CODE OPTIMIZATION TECHNIQUES CONSTANT FOLDING

AIM:

To write a C program to implement Constant Folding (Code optimization Technique).

ALGORITHM:

- The desired header files are declared.
- The two file pointers are initialized one for reading the C program from the file and one for writing the converted program with constant folding.
- The file is read and checked if there are any digits or operands present.
- If there is, then the evaluations are to be computed in switch case and stored.
- Copy the stored data to another file.
- Print the copied data file.

PROGRAM:

```
#include<stdio.h>
#include<string.h>
void main() {
       char s[20];
       char flag[20]="//Constant";
       char result, equal, operator;
       double op1,op2,interrslt;
       int a,flag2=0;
       FILE *fp1,*fp2;
       fp1 = fopen("input.txt","r");
       fp2 = fopen("output.txt","w");
       fscanf(fp1,"%s",s);
       while(!feof(fp1)) {
               if(strcmp(s,flag)==0) {
                       flag2 = 1;
               if(flag2==1) {
                       fscanf(fp1,"%s",s);
                       result=s[0];
                       equal=s[1];
                       if(isdigit(s[2])&& isdigit(s[4])) {
                               if(s[3]=='+'||'-'||'*'||'/')
                                       operator=s[3];
                                       switch(operator) {
                                               case '+':
                                                      interrslt=(s[2]-48)+(s[4]-48);
                                                      break;
                                               case '-':
                                                      interrslt=(s[2]-48)-(s[4]-48);
```

```
break;
                                               case '*':
                                                       interrslt=(s[2]-48)*(s[4]-48);
                                                       break;
                                               case '/':
                                                       interrslt=(s[2]-48)/(s[4]-48);
                                                       break;
                                               default:
                                                       interrslt = 0;
                                                       break; }
                                       fprintf(fp2,"/*Constant Folding*/\n");
                                       fprintf(fp2,"\%c = \%lf\n",result,interrslt);
                                       flag2 = 0;
                        } else {
                               fprintf(fp2,"Not Optimized\n");
                               fprintf(fp2, "%s\n",s);
                } else {
                       fprintf(fp2,"%s\n",s);
               fscanf(fp1,"%s",s);
       fclose(fp1);
       fclose(fp2);
}
```

OUTPUT:

```
[root@localhost-live 518_ex9]# vi input.txt
[root@localhost-live 518_ex9]# vi 518_ex9.c
[root@localhost-live 518_ex9]# cc 518_ex9.c
[root@localhost-live 518_ex9]# ./a.out
[root@localhost-live 518_ex9]# vi output.txt
```

//output.txt

```
a=7
b=10
c=5
d=7
```

RESULT: