Name: Gokulnath M Prabhu
Class: CS7B
Roll No: 21

Lab Cycle 4 - Experiment 17

Write a program to perform constant propagation

Code:

```c
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <ctype.h>

void input();
void output();
void change(int p, char *res);
void constant();

struct expr
{
   char op[2], op1[5], op2[5], res[5];
   int flag;
} arr[10];

int n;

void main()
{
   input();
   constant();
   output();
   getchar();
}

void input()
{
   int i;
   printf("Enter the maximum number of expressions : ");
   scanf("%d", &n);
   printf("\nEnter the input : \n");
   for (i = 0; i < n; i++)
   {
       scanf("%s", arr[i].op);
       scanf("%s", arr[i].op1);
       scanf("%s", arr[i].op2);
```

```
        scanf("%s", arr[i].res);
        arr[i].flag = 0;
    }
}

void constant()
{
    int i;
    int op1, op2, res;
    char op, res1[5];
    for (i = 0; i < n; i++)
    {
        if (isdigit(arr[i].op1[0]) && isdigit(arr[i].op2[0]) ||
strcmp(arr[i].op, "=") == 0) /*if both digits,
            store them in variables*/
        {
            op1 = atoi(arr[i].op1);
            op2 = atoi(arr[i].op2);
            op = arr[i].op[0];
            switch (op)
            {
            case '+':
                res = op1 + op2;
                break;
            case '-':
                res = op1 - op2;
                break;
            case '*':
                res = op1 * op2;
                break;
            case '/':
                res = op1 / op2;
                break;
            case '=':
                res = op1;
                break;
            }
            sprintf(res1, "%d", res);
            arr[i].flag = 1; /*eliminate expr and replace any operand
below that uses result of this expr */
            change(i, res1);
        }
    }
```

```
}

void output()
{
   int i = 0;
   printf("\nOptimized code is : ");
   for (i = 0; i < n; i++)
   {
       if (!arr[i].flag)
       {
           printf("\n%s %s %s %s", arr[i].op, arr[i].op1, arr[i].op2,
arr[i].res);
       }
   }
}

void change(int p, char *res)
{
   int i;
   for (i = p + 1; i < n; i++)
   {
       if (strcmp(arr[p].res, arr[i].op1) == 0)
           strcpy(arr[i].op1, res);
       else if (strcmp(arr[p].res, arr[i].op2) == 0)
           strcpy(arr[i].op2, res);
   }
}
```

Output:

```
➜  Constant_Propagation git:(master) ✗ gcc constant propagation.c
➜  Constant_Propagation git:(master) ✗ ./a.out
Enter the maximum number of expressions : 4

Enter the input :
= 3 - a
+ a b t1
+ a c t2
+ t1 t2 t3

Optimized code is :
+ 3 b t1
+ 3 c t2
+ t1 t2 t3
➜  Constant_Propagation git:(master) ✗
```