

Experiment 10: Constant Propagation

Aim: write a program to perform Constant propagation.

Theory:

Constants assigned to a Variable can be propagated through the flow graph and substituted at the use of variable.

eg $x = 3$

$y = x + 4$

Then, $y = 3 + 4$

Algorithm:

1) Start.

2) Read the instruction from input file

3) if length of string after = operation is 1 then copy L value & R value to array Constant & value.

4) else check if both operands are digit if yes print it.

5) else if one operand is digit & other is char then find the character in constant array & substitute the value.

6) else if both operands are not digit find both operands in const. array & replace value and print instruction.

7) stop

Result: The program was compiled successfully & the o/p was verified.

Program:

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

struct op
{
    char l;
    char r[20];
} op[10], pr[10];

int n;
char id[4] = {'+', '-', '*', '/'};

int check(char ch)
{
    for (int i = 0; i < 4; i++)
        if (ch == id[i])
            return 1;
    return 0;
}

int opcount(char str[])
{
    int count = 0;
    for (int i = 0; i < strlen(str); i++)
    {
        for (int j = 0; j < 4; j++)
        {
            if (str[i] == id[j])
                count++;
        }
    }
    return count;
}
```

```

int checknot(char ch, int n)
{
    for (int i = 0; i < n; i++)
        if (ch == op[i].l)
            return 0;
    return 1;
}

```

```

void prop(char str[], int n)
{
    if (opcount(str) == 0)
    {
        if (isdigit(str[0]))
        {
            printf("%s", str);
        }
        else
        {
            for (int i = 0; i < n; i++)
            {
                if (op[i].l == str[0])
                    prop(op[i].r, n);
            }
        }
    }
}

```

```

void main()
{
    int len = 0, n;
    char production[20];
    printf("\nEnter the Number of Productions : ");
    scanf("%d", &n);
    printf("Enter the Productions :\n");
    for (int i = 0; i < n; i++)
    {
        scanf(" %s", &production);
        len = strlen(production);
    }
}

```

```

    op[i].l = production[0];
    memcpy(op[i].r, &production[2], len - 1);
}
printf("\nAfter Code Propagation :-\n");
printf("-----");
for (int i = 0; i < n; i++)
{
    if (opcount(op[i].r) > 0)
    {
        printf("%c=", op[i].l);
        for (int j = 0; j < strlen(op[i].r); j++)
        {
            if (isdigit(op[i].r[j]) || check(op[i].r[j]) || checknot(op[i].r[j], n))
                printf("%c", op[i].r[j]);
            else
            {
                char str[2];
                str[0] = op[i].r[j];
                str[1] = '\0';
                prop(str, n);
            }
        }
    }
}
printf("\n");
}
}

```

Output:

```
Enter the Number of Productions : 4
```

```
Enter the Productions :
```

```
a=5
```

```
b=a+3
```

```
c=a
```

```
d=c+2
```

```
After Code Propagation :-
```

```
-----
```

```
b=5+3
```

```
d=5+2
```