

1. 14. Write a C Program for code optimization to eliminate common subexpression.

```
#include <stdio.h>

#include <conio.h>

#include <string.h>

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

int main() {
    int a, i, k, j, n, z = 0, m, q;
    char *p, *l;
    char temp, t;
    char *tem;

    printf("Enter the Number of Values:");
    scanf("%d", &n);
    for (i = 0; i < n; i++) {
        printf("left: ");
        op[i].l = getche();
        printf("\tright: ");
        scanf("%s", op[i].r);
    }
    printf("\nIntermediate Code\n");
    for (i = 0; i < n; i++) {
        printf("%c=%s\n", op[i].l, op[i].r);
    }
}
```

```

for (i = 0; i < n - 1; i++) {
    temp = op[i].l;
    for (j = 0; j < n; j++) {
        p = strchr(op[j].r, temp);
        if (p) {
            pr[z].l = op[i].l;
            strcpy(pr[z].r, op[i].r);
            z++;
        }
    }
}

pr[z].l = op[n - 1].l;
strcpy(pr[z].r, op[n - 1].r);
z++;

printf("\nAfter Dead Code Elimination\n");
for (k = 0; k < z; k++) {
    printf("%c=%s\n", pr[k].l, pr[k].r);
}

for (m = 0; m < z; m++) {
    tem = pr[m].r;
    for (j = m + 1; j < z; j++) {
        p = strstr(tem, pr[j].r);
        if (p) {
            t = pr[j].l;
            pr[j].l = pr[m].l;
            for (i = 0; i < z; i++) {
                l = strchr(pr[i].r, t);
                if (l) {
                    a = l - pr[i].r;

```

```

        printf("pos: %d", a);

        pr[i].r[a] = pr[m].l;

    }

}

}

}

}

printf("\nEliminate Common Expression\n");
for (i = 0; i < z; i++) {

    printf("%c=%s\n", pr[i].l, pr[i].r);

}

for (i = 0; i < z; i++) {

    for (j = i + 1; j < z; j++) {

        q = strcmp(pr[i].r, pr[j].r);

        if ((pr[i].l == pr[j].l) && !q) {

            pr[i].l = '\0';

            strcpy(pr[i].r, "\0");

        }

    }

}

printf("\nOptimized Code\n");
for (i = 0; i < z; i++) {

    if (pr[i].l != '\0') {

        printf("%c=%s\n", pr[i].l, pr[i].r);

    }

}

getch();

return 0;

}

```

E:\c program\Untitled1.exe

```
Enter the Number of Values:5
left: a right: 9
left: b right: c+d
left: e right: c+d
left: f right: b+e
left: r right: f
```

Intermediate Code

```
a=9
b=c+d
e=c+d
f=b+e
r=f
```

After Dead Code Elimination

```
b=c+d
e=c+d
f=b+e
r=f
```

pos: 2

Eliminate Common Expression

```
b=c+d
b=c+d
f=b+b
r=f
```

Optimized Code

```
b=c+d
f=b+b
r=f
```