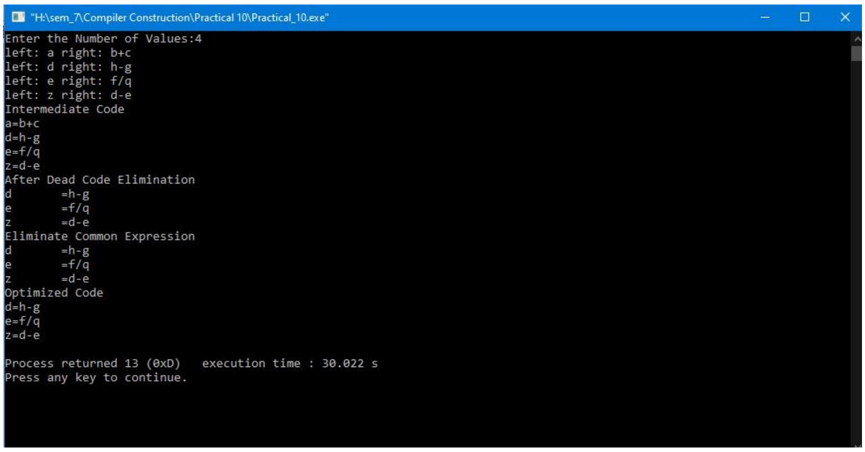
Name: Niketkumar Kothari

Roll No: 18BCE134

Practical 10 Title: Code Optimization

Techniques

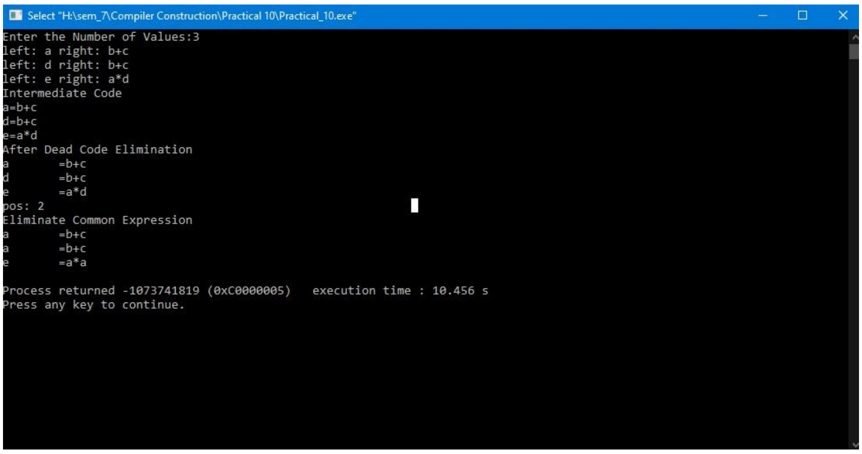
Output:



"H:\sem_7\Compiler Construction\Practical 10\Practical_10.exe"

```
Enter the Number of Values:4
left: a right: b+c
left: d right: h-g
left: e right: f/q
left: z right: d-e
Intermediate Code
a=b+c
d=h-g
e=f/q
z=d-e
After Dead Code Elimination
d       =h-g
e       =f/q
z       =d-e
Eliminate Common Expression
d       =h-g
e       =f/q
z       =d-e
Optimized Code
d=h-g
e=f/q
z=d-e

Process returned 13 (0xD)   execution time : 30.022 s
Press any key to continue.
```



Select "H:\sem_7\Compiler Construction\Practical 10\Practical_10.exe"

```
Enter the Number of Values:3
left: a right: b+c
left: d right: b+c
left: e right: a*d
Intermediate Code
a=b+c
d=b+c
e=a*d
After Dead Code Elimination
a       =b+c
d       =b+c
e       =a*d
pos: 2
Eliminate Common Expression
a       =b+c
a       =b+c
e       =a*a

Process returned -1073741819 (0xC0000005)   execution time : 10.456 s
Press any key to continue.
```

Code:

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
struct op { char l;
char r[20];
}
op[10],pr[10]; void
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("Intermediate Code\n") ;
for(i=0;i<n;i++)
{
printf("%c=",op[i].l);
printf("%s\n",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("After Dead Code Elimination\n"); for(k=0;k<z;k++) { printf("%c\t=",pr[k].l);
printf("%s\n",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\n",a);
pr[i].r[a]=pr[m].l; }}}}}
printf("Eliminate Common Expression\n"); for(i=0;i<z;i++)
{
printf("%c\t=",pr[i].l);
printf("%s\n",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("Optimized Code\n"); for(i=0;i<z;i++)
{ if(pr[i].l!='\0')
{
printf("%c=",pr[i].l); printf("%s\n",pr[i].r);
}
}
```

```
getch();
}
```