**Exercise 13**

**RA1911026010045**

**K1**

**Implement control flow analysis**

**AIM:**To write a C program to implement control Flow Analysis.

**ALGORITHM**

* Start the Program Execution
* Read the total Numbers of Expression
* Read the Left and Right side of Each Expressions
* Display the Expressions with Line No
* Display the Data flow movement with Particular Expressions
* Stop the Program Execution

**CODE:**

#include<stdio.h>

#include<string.h>

#include<ctype.h>

void input();

void output();

void change(int p,int q,char \*res);

void constant();

void expression();

struct expr

{

char op[2],op1[5],op2[5],res[5];

int flag;

}arr[10];

int n;

int main()

{

int ch=0;

input();

constant();

expression();

output();

}

void input()

{

int i;

printf("\n\nEnter 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]))

{

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;

}

sprintf(res1,"%d",res);

arr[i].flag=1;

change(i,i,res1);

}

}

}

void expression()

{

int i,j;

for(i=0;i<n;i++)

{

for(j=i+1;j<n;j++)

{

if(strcmp(arr[i].op,arr[j].op)==0)

{

if(strcmp(arr[i].op,"+")==0||strcmp(arr[i].op,"\*")==0)

{

if(strcmp(arr[i].op1,arr[j].op1)==0&&strcmp(arr[i].op2,arr[j].op2)==0 ||

strcmp(arr[i].op1,arr[j].op2)==0&&strcmp(arr[i].op2,arr[j].op1)==0)

{

arr[j].flag=1;

change(i,j,NULL);

}

}

else

{

if(strcmp(arr[i].op1,arr[j].op1)==0&&strcmp(arr[i].op2,arr[j].op2)==0)

{

arr[j].flag=1;

change(i,j,NULL);

}          }

}          }

}          }

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\n",arr[i].op,arr[i].op1,arr[i].op2,arr[i].res);

}

}

}

void change(int p,int q,char \*res)

{

int i;

for(i=q+1;i<n;i++)

{

if(strcmp(arr[q].res,arr[i].op1)==0)

if(res == NULL)

strcpy(arr[i].op1,arr[p].res);

else

strcpy(arr[i].op1,res);

else if(strcmp(arr[q].res,arr[i].op2)==0)

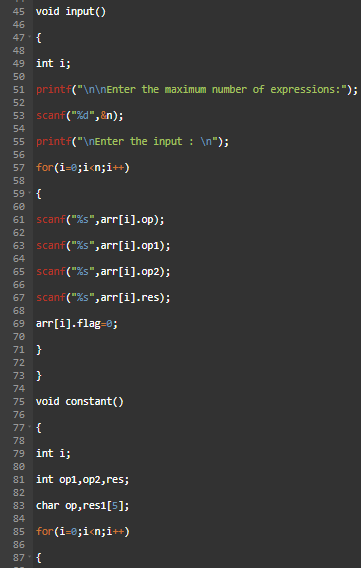
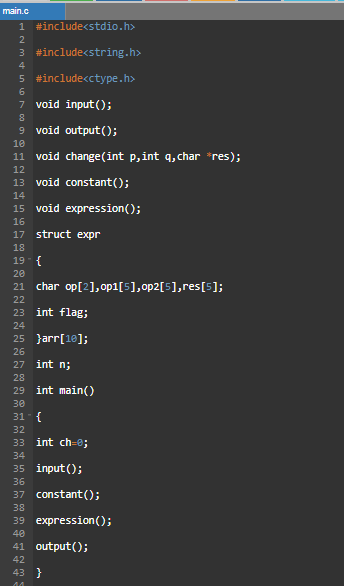
if(res == NULL)

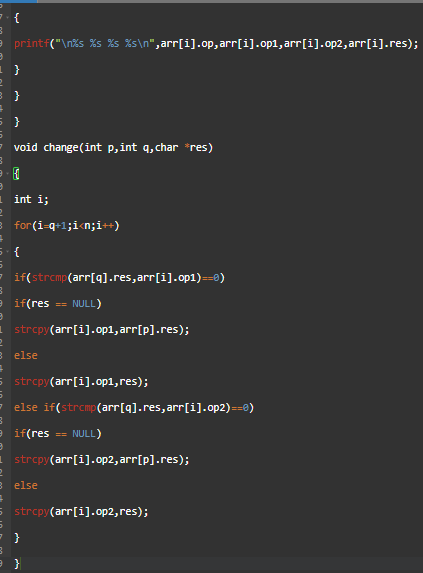
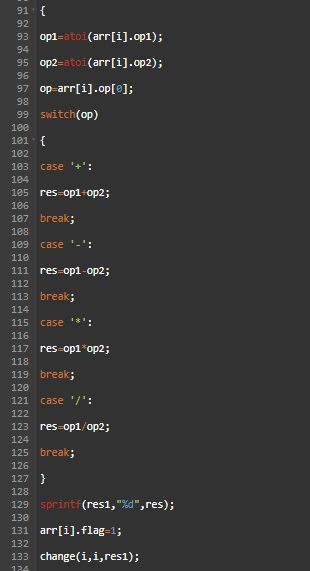
strcpy(arr[i].op2,arr[p].res);

else

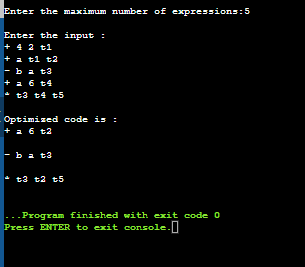
strcpy(arr[i].op2,res);}}

**CODE SCREENSHOTS:**

****

****

**OUTPUT:**

****

**Implement Data flow analysis**

**AIM:**To write a C program to implement Data Flow Analysis.

**ALGORITHM**

Step-1: Start the Program Execution

Step-2: Read the total Numbers of Expression

Step-3: Read the Left and Right side of Each Expressions

Step-4: Display the Expressions with Line No

Step-5: Display the Data flow movement with Particular Expressions

Step-6: Stop the Program Execution

**CODE:**

#include<stdio.h>

#include<conio.h>

#include <string.h>

struct op

{

char l[20];

char r[20];

}

op[10], pr[10];

void main()

{

int a, i, k, j, n, z = 0, m, q,lineno=1;

char \* p, \* l;

char temp, t;

char \* tem;char \*match;

printf("enter no of values");

scanf("%d",&n);

for (i = 0; i< n; i++)

{

printf("\tleft\t");

scanf("%s",op[i].l);

printf("\tright:\t");

scanf("%s", op[i].r);

}

printf("intermediate Code\n");

for (i = 0; i < n; i++)

{

printf("Line No=%d\n",lineno);

printf("\t\t\t%s=", op[i].l);

printf("%s\n", op[i].r);lineno++;

}

printf("\*\*\*Data Flow Analysis for the Above Code \*\*\*\n");

for(i=0;i<n;i++)

{

for(j=0;j<n;j++)

{

match=strstr(op[j].r,op[i].l);

if(match)

{

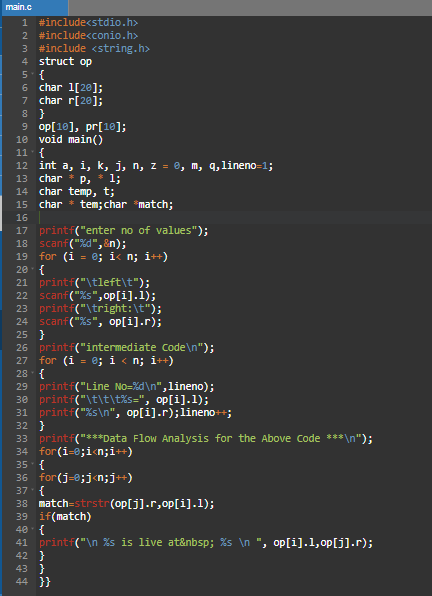
printf("\n %s is live at&nbsp; %s \n ", op[i].l,op[j].r);

}

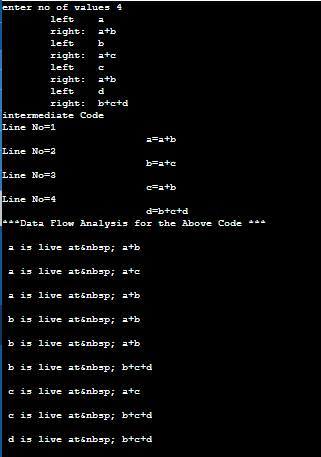
}

}}

**CODE SCREENSHOTS:**



**OUTPUT:**



**RESULT:**

Thus implementation of control flow analysis and data SLO-2 flow analysis was executed successfully.