1**.ALGORITHM:**

//Simple calculator

//The concepts used to implement this program are of pointer to function

1. Functions for addition, subtraction, multiplication, division and

display are initialised and declared.

2. The return type of these functions can be float or int.

3. In the main function

a. Declare the two numbers, answer variable, character for

choice.

b. Declare two function pointers (one for float type and

another for void).

c Take the input variable values (two numbers and choice).

d. Start switch statement with the choice.

e. Assign the pointer to the fuction name based on the

choice made.

f. Store the returned value in variable of answer.

g. display the answer using second pointer.

h. Use of goto or while statement can help to loop the

program.

**2.CODE WITH COMMENTS:**

**A]** #include<stdio.h>

void display(float s);

float add(int x,int y);

float subtract(int x,int y);

float multiply(int x,int y);

float divide(int x,int y);

//function definition

void display(float s)

{

printf("\nAns is:%f\n",s);

}

float add(int x,int y)

{

return x+y;

}

float subtract(int x,int y)

{

return x-y;

}

float multiply(int x,int y)

{

return x\*y;

}

float divide(int x,int y)

{

if(y!=0)

return (float)x/y;

else

return 0.0;

}

int main()

{

int a,b;

float ans;

unsigned char ch;

printf("Enter the two numbers separating with the single space:");

scanf("%d%d",&a,&b);

printf("Enter the arithmetic operation(+,-,\* and /):");

scanf(" %c",&ch);

switch(ch)

{

//calling add function

case'+':ans=add(a,b);

break;

//calling subtract function

case'-':ans=subtract(a,b);

break;

//calling multiply function

case'\*':ans=multiply(a,b);

break;

//calling divide function

case'/':ans=divide(a,b);

break;

default:printf("Invalid operation");

break;

}

//calling display function

display(ans);

return 0;

}

**B]**

#include<stdio.h>

void display(float s)

{

printf("\nAns is:%f\n",s);

}

float add(int x,int y)

{

return x+y;

}

float subtract(int x,int y)

{

return x-y;

}

float multiply(int x,int y)

{

return x\*y;

}

float divide(int x,int y)

{

return x/y;

}

int main()

{

int a,b;

float ans;

unsigned char ch;

float(\*ptr\_op)(int,int);

//pointer to function declaration

float(\*ptr\_disp)(float);

ptr\_disp=display;

//pointer initialization

printf("enter the two numbers separating with space=",a,b);

scanf("%d %d",&a,&b);

printf("enter the arithmetic operation(+,-,\* or /):");

scanf(" %c",&ch);

switch(ch)

{

case'+':ptr\_op=add;

break;

//add function assigned to pointer

case'-':ptr\_op=subtract;

break;

//subtract function assigned to pointer

case'\*':ptr\_op=multiply;

break;

//multiply function assigned to pointer

case'/':ptr\_op=divide;

break;

//divide function assigned to pointer

default:printf("invalid operation");

break;

}

ans=(\*ptr\_op)(a,b);

//function call using function pointer to perform the operations

(\*ptr\_disp)(ans);

return 0;

}

**3.SAMPLE OUTPUT INPUT OF BOTH A AND B TYPE CODES**

enter the two numbers separating with space=30 5

enter the arithmetic operation(+,-,\* or /):+

Ans is:35.000000

enter the two numbers separating with space=30 5

enter the arithmetic operation(+,-,\* or /):-

Ans is:25.000000

enter the two numbers separating with space=30 5

enter the arithmetic operation(+,-,\* or /):\*

Ans is:150.000000

enter the two numbers separating with space=30 5

enter the arithmetic operation(+,-,\* or /):/

Ans is:6.000000

enter the two numbers separating with space=\_

enter the arithmetic operation(+,-,\* or /):invalid operation