EXPERIMENT13: MACROS IN C

Experiment 1: Arithmetic Operations Using Macros

Aim

To define and use multiple macros in C to perform basic arithmetic operations.

Algorithm

1. Define macros for addition, subtraction, multiplication, and division
2. Read two numbers
3. Apply macro operations
4. Display results

Pseudocode

```
DEFINE ADD(a,b)
DEFINE SUB(a,b)
DEFINE MUL(a,b)
DEFINE DIV(a,b)

READ a, b
```

```
DISPLAY ADD, SUB, MUL, DIV
```

Flowchart (ASCII)

```
Start
   ↓
Define Macros
   ↓
Read Two Numbers
   ↓
Apply Macros
   ↓
Display Results
   ↓
 End
```

C Program

```
#include <stdio.h>

#define ADD(a, b) ((a) + (b))
#define SUB(a, b) ((a) - (b))
#define MUL(a, b) ((a) * (b))
#define DIV(a, b) ((a) / (b))

int main() {
    int x, y;

    printf("Enter two integers: ");
    scanf("%d %d", &x, &y);

    printf("Addition = %d\n", ADD(x, y));
    printf("Subtraction = %d\n", SUB(x, y));
```

```
    printf("Multiplication = %d\n", MUL(x, y));

    if (y != 0)
        printf("Division = %d\n", DIV(x, y));
    else
        printf("Division by zero not allowed\n");

    return 0;
}
```

Output

```
PS C:\Users\ASUS\Desktop\Dahadi\class c> gcc t.c
PS C:\Users\ASUS\Desktop\Dahadi\class c> ./a.exe
Enter two integers: 2
3
Addition = 5
Subtraction = -1
Multiplication = 6
Division = 0
PS C:\Users\ASUS\Desktop\Dahadi\class c>
```

Conclusion

Macros allow fast execution of arithmetic operations by avoiding function call overhead. Proper use of parentheses prevents logical errors.