

EXPERIMENT3.2 – LOOPS

Q1. Program to enter numbers till user wants & display count of +ve, -ve, zero

Aim

To repeatedly accept numbers from the user until they choose to stop, and then print the count of positive, negative, and zero values.

Algorithm

1. Start.
2. Initialize counters: positive = 0, negative = 0, zero = 0.
3. Use a loop that continues until the user decides to stop.
4. Input a number.
5. If number > 0 → increment positive.
6. Else if number < 0 → increment negative.
7. Else → increment zero.
8. Ask the user whether they want to continue.
9. End loop when user enters 'n' or 'N'.
10. Display the counts.
11. Stop.

Pseudocode

```
pos = neg = zero = 0
repeat
    input num
    if num > 0 then pos++
    else if num < 0 then neg++
    else zero++
    ask user continue? (y/n)
until user enters n
print pos, neg, zero
```

Flowchart (Text)

```
START
↓
Initialize counters
↓
Repeat Loop:
↓
Input number
↓
Check sign (+ / - / 0)
↓
Update counter
↓
Ask: Continue? y/n
↓
If yes → repeat
If no → exit loop
↓
Print result
↓
STOP
```

C Program

```
#include <stdio.h>

int main() {
    int num, pos = 0, neg = 0, zero = 0;
    char ch;

    do {
        printf("Enter a number: ");
        scanf("%d", &num);

        if (num > 0) pos++;
        else if (num < 0) neg++;
        else zero++;

        printf("Do you want to continue? (y/n): ");
        scanf(" %c", &ch);

    } while (ch == 'y' || ch == 'Y');

    printf("\nPositive numbers: %d", pos);
    printf("\nNegative numbers: %d", neg);
    printf("\nZeroes: %d\n", zero);

    return 0;
}
```

Output

```
PS C:\Users\ASUS\Desktop\C Exp> gcc exp3.2.1.c
PS C:\Users\ASUS\Desktop\C Exp> ./a.exe
Enter a number: 4
Do you want to enter another number? (y/n): y
Enter a number: -5
Do you want to enter another number? (y/n): y
Enter a number: 0
Do you want to enter another number? (y/n): n

Count of Positive numbers = 1
Count of Negative numbers = 1
Count of Zeros = 1
PS C:\Users\ASUS\Desktop\C Exp> 
```

Q2. Print multiplication table of a number

Aim

To print the multiplication table of a number entered by the user.

Algorithm

1. Start.
2. Input the number.
3. Loop i from 1 to 10.
4. Print number \times i.
5. Stop.

Pseudocode

```
input n
for i = 1 to 10
    print n × i
```

Flowchart

START → Input number → Loop i = 1 to 10 → Print n*i → END

C Program

```
#include <stdio.h>

int main() {
    int n;
    printf("Enter a number: ");
    scanf("%d", &n);

    for (int i = 1; i <= 10; i++)
        printf("%d × %d = %d\n", n, i, n * i);

    return 0;
}
```

Output

```
Multiplication Table of 4
```

```
-----
```

```
4 x 1 = 4
```

```
4 x 2 = 8
```

```
4 x 3 = 12
```

```
4 x 4 = 16
```

```
4 x 5 = 20
```

```
4 x 6 = 24
```

```
4 x 7 = 28
```

```
4 x 8 = 32
```

```
4 x 9 = 36
```

```
4 x 10 = 40
```

```
PS C:\Users\ASUS\Desktop\C Exp>
```

Q3. Generate the following pattern

a)

```
*
```

```
**
```

```
***
```

```
****
```

b)

1
12
123
1234

Aim

To print custom star and number patterns using loops.

Algorithm

1. Start.
2. Loop through rows.
3. For each row, print required characters.
4. Stop.

Pseudocode

```
for r = 1 to n  
    print r stars
```

```
for r = 1 to n  
    print numbers from 1 to r
```

C Program

```
#include <stdio.h>  
  
int main() {  
    int i, j;  
  
    // Pattern A  
    for (i = 1; i <= 4; i++) {
```

```
        for (j = 1; j <= i; j++) {
            printf("*");
        }
        printf("\n");
    }

    // Pattern B
    for (i = 1; i <= 4; i++) {
        for (j = 1; j <= i; j++) {
            printf("%d", j);
        }
        printf("\n");
    }

    return 0;
}
```

Output

```
PS C:\Users\ASUS\Desktop\Dahadi\class c> gcc loop1.c
PS C:\Users\ASUS\Desktop\Dahadi\class c> ./a.exe
*
**
***
****
1
12
123
1234
PS C:\Users\ASUS\Desktop\Dahadi\class c> █
```

Q4. Town population problem

Initial population = 100000

Growth rate = 10% per year

Print population at end of each year for last 10 years

Aim

To compute the population of a town after each year using a fixed growth rate.

Algorithm

1. Start.
2. Initialize population = 100000.
3. Loop for 10 years:
 - a. $\text{population} = \text{population} + (\text{population} \times 0.10)$
 - b. print population
4. Stop.

Pseudocode

```
pop = 100000
for year = 1 to 10
    pop = pop + 0.10 * pop
    print pop
```

C Program

```
#include <stdio.h>
```

```
int main() {  
    float pop = 100000;  
  
    for (int year = 1; year <= 10; year++) {  
        pop = pop + (0.10 * pop);  
        printf("Year %d Population: %.2f\n", year, pop);  
    }  
  
    return 0;  
}
```

Output

```
PS C:\Users\ASUS\Desktop\C Exp> gcc exp3.2.4.c  
PS C:\Users\ASUS\Desktop\C Exp> ./a.exe  
Year      Population  
-----  
1          110000.00  
2          121000.00  
3          133100.00  
4          146410.00  
5          161051.00  
6          177156.09  
7          194871.70  
8          214358.88  
9          235794.77  
10         259374.25  
PS C:\Users\ASUS\Desktop\C Exp> 
```

Q5. Ramanujan Numbers up to a limit

A Ramanujan number satisfies:

$$a^3 + b^3 = c^3 + d^3 \quad (a, b, c, d > 0)$$

Aim

To print all Ramanujan (taxicab) numbers up to a given limit.

Algorithm

1. Start.
2. Input limit.
3. Use 4 nested loops to test combinations.
4. If $a^3 + b^3 == c^3 + d^3 \rightarrow$ print number.
5. Stop.

Pseudocode

```
input limit
for a=1 to limit
  for b=a to limit
    for c=1 to limit
      for d=c to limit
        if (a^3 + b^3 == c^3 + d^3)
          print number
```

C Program

```
#include <stdio.h>
```

```

int main() {
    int limit;
    printf("Enter limit: ");
    scanf("%d", &limit);

    for (int a = 1; a <= limit; a++)
        for (int b = a; b <= limit; b++)
            for (int c = 1; c <= limit; c++)
                for (int d = c; d <= limit; d++) {
                    int s1 = a*a*a + b*b*b;
                    int s2 = c*c*c + d*d*d;

                    if (s1 == s2 && a != c && a != d) {
                        printf("%d = %d^3 + %d^3 = %d^3 + %d^3\n",
                            s1, a, b, c, d);
                    }
                }
    return 0;
}

```

Output

```

PS C:\Users\ASUS\Desktop\C Exp> gcc exp3.2.5.c
PS C:\Users\ASUS\Desktop\C Exp> ./a.exe
Enter limit: 2000

Ramanujan Numbers up to 2000 are:
-----
1729 = 1^3 + 12^3 = 9^3 + 10^3
PS C:\Users\ASUS\Desktop\C Exp> 

```