

6. Decision making and looping

* The while statement

The simplest of all the looping structures in C is the while loop.

Syntax : while (test-condition)
{
 body of the loop
}

1) A program to evaluate the equation $y = x^n$.

```
main()
{
    int count, n;
    float x, y;
    printf("Enter the values of x and n: ");
    scanf("%f %d", &x, &n);
    y = 1.0;
    count = 1;
    while (count <= n)
    {
        y = y * x;
        count++;
    }
    printf("x = %f, n = %d, x to power n = %f", x, n, y);
}
```

* The do statement

do
{
 body of the loop
}
while (test-condition);

* The for statement

for (initialization; test condition; increment)
{
 body of the loop
}

2) Power of 2 table using for loop

```
main()
{
    long int p;
    int n;
    double q;
    printf("2 to power n\n");
    p = 1;
    for (n = 0; n <= 21; ++n)
```

```

if (n == 0)
    p = 1;

```

```

else
    p = p * 2;

```

```

q = 1.0 / (double) p;

```

```

printf("u: %10i d: %10i 20: %12i, f(n): %10i, p: %10i, q: %10f\n", n, p, n, q);

```

```

printf("-----\n");

```

```

}

```

3) Fibonacci number of nth

```

#include <stdio.h>

```

```

#include <conio.h>

```

```

void main()

```

```

{

```

```

    int num1 = 0, num2 = 1, n, i, ffb;

```

```

    clrscr();

```

```

    printf("Enter the value of n: ");

```

```

    scanf("%d", &n);

```

```

    for (i = 1; i <= n - 2; i++)
    {

```

```

        ffb = num1 + num2;

```

```

        num1 = num2;

```

```

        num2 = ffb;
    }

```

```

    printf("nth fibonacci number (for n = %d) = %d, n, ffb);

```

```

    getch();

```

```

}

```

4) display a pyramid.

```

#include <stdio.h>

```

```

#include <conio.h>

```

```

void main()

```

```

{

```

```

    int num, i, y, x = 40;

```

```

    clrscr();

```

```

    printf("Enter a number for generating the pyramid: ");

```

```

    scanf("%d", &num);

```

```

    for (y = 0; y <= num; y++)
    {

```

```

        {

```

```

            gotoxy(x, y + 1);

```

```

            supriya: for (i = 0 - y; i <= y; i++)

```

```
printf("%i, %d", abs(i));
```

```
a = 2-3;
```

```
}
```

```
getch();
```

```
}
```

* Solving a loop.

- Analyze the problem and see whether it required pre-test or post-test loop
- post test loop → do-while
- pre test loop → for and while loop.
- decide whether the loop terminates on counter based on sentinel based

counter based → for loop

sentinel based → while loop.

5) print all even numbers up to N

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

```
int main() {
```

```
int i, N;
```

```
printf("Enter N:");
```

```
scanf("%d", &N);
```

```
printf("Even numbers up to %d, N);
```

```
for (i = 1; i <= N; i++)
```

```
if (i % 2 == 0)
```

```
printf("%d ", i);
```

```
return 0;
```

```
}
```

6) palin drome or not

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

```
int main() {
```

```
int num, rev = 0, temp;
```

```
printf("Enter a number:");
```

```
scanf("%d", &num);
```

```
temp = num;
```

```
while (temp != 0) {
```

```
rev = rev * 10 + temp % 10;
```

```
temp = temp / 10;
```

```
}
```

```
if (num == rev)
```

```
printf("%d is a palindrome", num);
```

```
else
```

```
printf("%d is a not palindrome");
```

```
}
```

supriya ✨

7) Armstrong

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

```
#include <math.h>
```

```
int main() {
```

```
    int i, N, temp, sum, digit, ndigit;
```

```
    printf("Enter N: ");
```

```
    scanf("%d", &N);
```

```
    printf("Armstrong numbers up to %d", N);
```

```
    for (i = 1; i <= N; i++) {
```

```
        temp = i; sum = 0;
```

```
        ndigit = 0; int t = temp;
```

```
        while (t != 0) {
```

```
            t = t / 10;
```

```
            ndigit++;
```

```
        }
```

```
        while (temp != 0) {
```

```
            digit = temp % 10;
```

```
            sum = sum + pow(digit, ndigit);
```

```
            temp = temp / 10;
```

```
        }
```

```
        if (sum == i) printf("%d ", i);
```

```
    }
```

```
    return 0;
```

```
}
```

8) multiplication table

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

```
int main() {
```

```
    int num, i;
```

```
    printf("Enter a number: ");
```

```
    scanf("%d", &num);
```

```
    printf("multiplication table of %d is\n", num);
```

```
    for (i = 1; i <= 10; i++)
```

```
        printf("%d x %d = %d\n", num, i, num * i);
```

```
    return 0;
```

```
}
```

9. Triangle pattern of *

#include <stdio.h>

int main()

{ int i, j, n;

printf("Enter number of rows: ");

scanf("%d", &n);

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

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

{ printf(" ");

for (j=1; j<=2*i-1; j++)

{ printf("* ");

printf("\n");

}

return 0;

}

10. All prime numbers up to N.

#include <stdio.h>

int main()

{ int i, j, N, flag;

printf("Enter N: ");

scanf("%d", &N);

printf("Prime numbers up to %d:", N);

for (i=2; i<=N; i++)

{ flag=1;

for (j=2; j<=i/2; j++)

{ if (i%j==0)

{ flag=0;

break;

if (flag)

{ printf("%d ", i);

}

return 0;

}

```

#include <stdio.h>
int main() {
    long long int num;
    long long int sum = 0;
    int count = 0;
    for (int i = 0; i < 10; i++) {
        printf("Enter integer %d: ", i + 1);
        if (scanf("%lld", &num) != 1){
            printf("Invalid input. Exiting.\n");
            break;
        }
        if (num > 0) {
            sum += num;
            count++;
        }
        if (sum > 999) {
            break;
        }
    }
    printf("Sum of positive values: %lld\n", sum);
    printf("Number of positive values added: %d\n", count);
    return 0;
}

```

supriya@ubuntu:~/Desktop/c/chp6\$./goto

```

Enter integer 1: 34
Enter integer 2: 23
Enter integer 3: 12
Enter integer 4: 78
Enter integer 5: 67
Enter integer 6:

```

98

```

Enter integer 7: 10
Enter integer 8: 999
Sum of positive values: 1321
Number of positive values added: 8

```

```
#include <stdio.h>
int main() {
    int i, count = 0;
    printf("Integers between 1 and 100 not divisible by 2 or 3:\n");
    for (i = 1; i <= 100; i++) {
        if (i % 2 != 0 && i % 3 != 0) {
            printf("%d ", i);
            count++;
        }
    }
    printf("\nTotal count = %d\n", count);
    return 0;
}
```

```
supriya@ubuntu:~/Desktop/c/chp6$ ./count
Integers between 1 and 100 not divisible by 2 or 3:
1 5 7 11 13 17 19 23 25 29 31 35 37 41 43 47 49 53 55 59 61 65 67 71 73 77 79 8
3 85 89 91 95 97
Total count = 33
```



```

#include <stdio.h>
int main() {
    int i, j;
    printf("Pattern (a):\n");
    for (i = 1; i <= 5; i++) {
        for (j = 1; j <= 5; j++) {
            printf("S ");
        }
        printf("\n");
    }
    printf("\nPattern (b):\n");
    for (i = 1; i <= 5; i++) {
        for (j = 1; j <= 5; j++) {
            if (i == 1 || i == 5 || j == 1 || j == 5) {
                printf("S ");
            } else {
                printf("  ");
            }
        }
        printf("\n");
    }
    return 0;
}

```

supriya@ubuntu:~/Desktop/c/chp6\$./squ

Pattern (a):

```

S S S S S
S S S S S
S S S S S
S S S S S
S S S S S

```

Pattern (b):

```

S S S S S
S           S
S           S
S           S
S           S
S S S S S

```



```
#include <stdio.h>
int main() {
    int i, j;
    for (i = 1; i <= 5; i++) {
        for (j = 1; j <= 5; j++) {
            if (i == 3 && j == 3) {
                printf("0 ");
            } else {
                printf("S ");
            }
        }
        printf("\n");
    }
    return 0;
}
```

supriya@ubuntu:~/Desktop/c/chp6\$./o

```
S S S S S
S S S S S
S S 0 S S
S S S S S
S S S S S
```

```

#include <stdio.h>
#include <math.h>
int main() {
    double C, d, P;
    double n;
    printf("Enter the original cost (C): ");
    scanf("%lf", &C);
    printf("Enter the depreciation rate (d) (e.g., 0.1 for 10%%): ");
    scanf("%lf", &d);
    printf("Enter the scrap value (P): ");
    scanf("%lf", &P);
    if (C <= 0 || P <= 0 || d <= 0 || d >= 1) {
        printf("Invalid input values.\n");
        return 1;
    }
    n = log(P / C) / log(1 - d);
    printf("The useful life of the item is approximately %.2f years.\n", n);
    return 0;
}

```

```

supriya@ubuntu:~/Desktop/c/chp6$ ./item
Enter the original cost (C): 1000
Enter the depreciation rate (d) (e.g., 0.1 for 10%): 0.5
Enter the scrap value (P): 1500
The useful life of the item is approximately -0.58 years.

```

```
#include <stdio.h>
int main() {
    double e = 1.0, prev_e = 0.0;
    double term = 1.0;
    int n = 1;
    while ((e - prev_e) >= 0.00001) {
        prev_e = e;
        term = term / n;
        e = e + term;
        n++;
    }
    printf("Approximated value of e = %.5f\n", e);
    printf("Number of terms used = %d\n", n);
    return 0;
}
```

```
supriya@ubuntu:~/Desktop/c/chp6$ ./euler
Approximated value of e = 2.71828
Number of terms used = 10
```



```

#include <stdio.h>
int main() {
    int num, i = 0;
    int binary[32];
    printf("Enter a positive integer: ");
    scanf("%d", &num);
    if (num <= 0) {
        printf("Please enter a positive integer.\n");
        return 1;
    }
    while (num > 0) {
        binary[i] = num % 2;
        num = num / 2;
        i++;
    }

    printf("Binary equivalent: ");
    for (int j = i - 1; j >= 0; j--) {
        printf("%d", binary[j]);
    }
    printf("\n");
    return 0;
}

```

```

supriya@ubuntu:~/Desktop/c/chp6$ ./binary
Enter a positive integer: 6
Binary equivalent: 110

```

```
#include <stdio.h>
int main() {
    int num, reversed = 0, digit;
    printf("Enter a number: ");
    scanf("%d", &num);
    int original = num;
    while (num > 0) {
        digit = num % 10;
        reversed = reversed * 10 + digit;
        num = num / 10;
    }
    printf("Original number: %d\n", original);
    printf("Reversed number: %d\n", reversed);
    return 0;
}
```

```
supriya@ubuntu:~/Desktop/c/chp6$ ./digit
```

```
Enter a number: 45678
```

```
Original number: 45678
```

```
Reversed number: 87654
```

```

#include <stdio.h>
int main() {
    int m, i;
    unsigned long long fact;
    printf("Enter a positive integer m: ");
    scanf("%d", &m);
    if (m < 0) {
        printf("Factorial is not defined for negative numbers.\n");
        return 1;
    }
    printf("Factorial table:\n");
    for (i = 1; i <= m; i++) {
        fact = 1;
        for (int j = 1; j <= i; j++) {
            fact *= j;
        }
        printf("%d! = %llu\n", i, fact);
    }
    return 0;
}

```

```

supriya@ubuntu:~/Desktop/c/chp6$ ./fac

```

```

Enter a positive integer m: 5

```

```

Factorial table:

```

```

1! = 1

```

```

2! = 2

```

```

3! = 6

```

```

4! = 24

```

```

5! = 120

```



```
#include <stdio.h>
int main() {
    int num, sum = 0, digit;
    printf("Enter an integer: ");
    scanf("%d", &num);
    int original = num;
    if (num < 0) {
        num = -num;
    }
    while (num > 0) {
        digit = num % 10;
        sum += digit;
        num = num / 10;
    }
    printf("Sum of digits of %d = %d\n", original, sum);
    return 0;
}
```

```
supriya@ubuntu:~/Desktop/c/chp6$ ./sum
Enter an integer: 342156
Sum of digits of 342156 = 21
```

```

#include <stdio.h>
int main() {
    int num, reversed = 0;
    printf("Enter a positive integer: ");
    scanf("%d", &num);
    if (num < 0) {
        printf("Please enter a positive integer.\n");
        return 1;
    }
    int temp = num;
    int digits = 0;
    for (; temp > 0; temp /= 10) {
        digits++;
    }
    temp = num;
    for (int i = 0; i < digits; i++) {
        int lastDigit = temp % 10;
        reversed = reversed * 10 + lastDigit;
        temp = temp / 10;
    }
    printf("Reversed number: %d\n", reversed);
    return 0;
}

```

```

supriya@ubuntu:~/Desktop/c/chp6$ ./for
Enter a positive integer: 56789
Reversed number: 98765

```