Practical File

Of

Programming in

Course Code: CSEG1041
School of Computer Science

Submitted By: Submitted To:
DYUTI SHARMA DR. PIYUSH BAGLA

Student Name: Dyuti Sharma
SAP ID:590021983
Course:B.Sc(Computer Science)
Batch:2025-28
Academic Year: 2025-26

```c
// File: 4.3.3_pattern.c
#include <stdio.h>
#define MAX_ROWS 5

/*
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
*/
int main() {
    int i, j;
    int current_number = 1;
    for (i = 1; i <= MAX_ROWS; i++) {
        for (j = 1; j <= i; j++) {
            printf("%d ", current_number++);
        }
        printf("\n");
    }
    return 0;
}
```

OUTPUT:

```
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
Program ended with exit code: 0
```

```c
// File: 4.3.2_pattern.c
#include <stdio.h>
#define MAX_ROWS 5

/*
1
12
123
1234
12345
*/
int main() {
    int i, j;
    for (i = 1; i <= MAX_ROWS; i++) {
        for (j = 1; j <= i; j++) {
            printf("%d", j);
        }
        printf("\n");
    }
    return 0;
}
```

OUTPUT:

```
1
12
123
1234
12345
Program ended with exit code: 0
```

```c
// File: 4.3.3_pattern.c
#include <stdio.h>
#define MAX_ROWS 5

/*
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
*/
int main() {
    int i, j;
    int current_number = 1;
    for (i = 1; i <= MAX_ROWS; i++) {
        for (j = 1; j <= i; j++) {
            printf("%d ", current_number++);
        }
        printf("\n");
    }
    return 0;
}
```

OUTPUT:

```
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
Program ended with exit code: 0
```

```c
// File: 4.3.4_pattern.c
#include <stdio.h>
#define MAX_ROWS 5

/*
1
21
321
4321
54321
*/
int main() {
    int i, j;
    for (i = 1; i <= MAX_ROWS; i++) {
        for (j = i; j >= 1; j--) {
            printf("%d", j);
        }
        printf("\n");
    }
    return 0;
}
```

OUTPUT:

```
1
21
321
4321
54321
Program ended with exit code: 0
```

```c
// File: 4.3.5_pattern.c
#include <stdio.h>
#define MAX_ROWS 5

/*
54321
5432
543
54
5
*/
int main() {
    int i, j;
    for (i = 1; i <= MAX_ROWS; i++) {
        for (j = MAX_ROWS; j >= i; j--) {
            printf("%d", j);
        }
        printf("\n");
    }
    return 0;
}
```

OUTPUT:

```
54321
5432
543
54
5
Program ended with exit code: 0
```

```c
// File: 4.3.6_pattern.c
#include <stdio.h>
#define MAX_ROWS 5

/*
    1
   12
  123
 1234
12345
*/
int main() {
    int i, j, s;
    for (i = 1; i <= MAX_ROWS; i++) {
        // Print spaces for alignment
        for (s = 1; s <= (MAX_ROWS - i); s++) {
            printf(" ");
        }
        // Print numbers (1 to i)
        for (j = 1; j <= i; j++) {
            printf("%d", j);
        }
        printf("\n");
    }
    return 0;
}
```

OUTPUT:

```
54321
5432
543
54
5
Program ended with exit code: 0
```

```c
// File: 4.3.7_pattern.c
#include <stdio.h>
#define MAX_ROWS 5

/*
12345
1234
123
12
1
*/
int main() {
    int i, j;
    for (i = MAX_ROWS; i >= 1; i--) {
        for (j = 1; j <= i; j++) {
            printf("%d", j);
        }
        printf("\n");
    }
    return 0;
}
```

OUTPUT:

```
12345
1234
123
12
1
Program ended with exit code: 0
```

```c
// File: 4.3.8_pattern.c
#include <stdio.h>
#define MAX_DIAMOND_ROWS 5

/*
        1
      1 2 1
    1 2 3 2 1
  1 2 3 4 3 2 1
1 2 3 4 5 4 3 2 1
  1 2 3 4 3 2 1
    1 2 3 2 1
      1 2 1
        1
*/
int main() {
    int i, j, s, k;
    int rows = MAX_DIAMOND_ROWS;

    // Upper Half
    for (i = 1; i <= rows; i++) {
        for (s = 1; s <= rows - i; s++) { printf("  "); }
        for (j = 1; j <= i; j++) { printf("%d ", j); }
        for (k = i - 1; k >= 1; k--) { printf("%d ", k); }
        printf("\n");
    }

    // Lower Half
    for (i = rows - 1; i >= 1; i--) {
        for (s = 1; s <= rows - i; s++) { printf("  "); }
        for (j = 1; j <= i; j++) { printf("%d ", j); }
        for (k = i - 1; k >= 1; k--) { printf("%d ", k); }
        printf("\n");
    }
    return 0;
}
```

OUTPUT:

```
        1
      1 2 1
    1 2 3 2 1
  1 2 3 4 3 2 1
1 2 3 4 5 4 3 2 1
  1 2 3 4 3 2 1
    1 2 3 2 1
      1 2 1
        1
Program ended with exit code: 0
```

```c
// File: 4.3.9_pattern.c
#include <stdio.h>
#define MAX_DIAMOND_ROWS 5

/*
5
5 4 5
5 4 3 4 5
5 4 3 2 3 4 5
5 4 3 2 1 2 3 4 5
5 4 3 2 3 4 5
5 4 3 4 5
5 4 5
5
*/
int main() {
    int i, j, s, k;
    int rows = MAX_DIAMOND_ROWS;

    // Upper Half (Inverted Pyramid)
    for (i = 1; i <= rows; i++) {
        for (s = 1; s < i; s++) { printf("  "); }
        for (j = rows; j >= i; j--) { printf("%d ", j); }
        for (k = i + 1; k <= rows; k++) { printf("%d ", k); }
        printf("\n");
    }

    // Lower Half (Pyramid)
    for (i = 2; i <= rows; i++) {
        for (s = 1; s < rows - i + 1; s++) { printf("  "); }
        for (j = rows; j >= i; j--) { printf("%d ", j); }
        for (k = i + 1; k <= rows; k++) { printf("%d ", k); }
        printf("\n");
    }
    return 0;
}
```

OUTPUT:

```
5 4 3 2 1 2 3 4 5
  5 4 3 2 3 4 5
    5 4 3 4 5
      5 4 5
        5
      5 4 3 2 3 4 5
    5 4 3 4 5
  5 4 5
5
Program ended with exit code: 0
```