Assignment 5

*Program 1: Half pyramid of **

ALGORITHM:

1. Accept an input value for number of rows from user.
2. Store the value in a variable (rows).
3. Declare two variables a and b.
4. Write an outer for loop with variable a, and a nested for loop containing the variable b.
5. Execution continues till outer loop condition is false.

CODE:

```
#include <stdio.h>
int main()
{
   int a, b, rows;
   printf("Enter the number of rows: ");
   scanf("%d", &rows);

   for (a = 1; a <= rows; ++a)
   {
      for (b = 1; b <= a; ++b)
      {
         printf("*");
      }
      printf("\n");
   }
   return 0;
```

OUTPUT:

```
Enter the number of rows: 9
```

```
*
* *
* * *
* * * *
* * * * *
* * * * * *
* * * * * * *
* * * * * * * *
* * * * * * * * *
```

Program 2: Half pyramid of numbers

ALGORITHM:

1. Accept an input value for number of rows from user.
2. Store the value in a variable (rows).
3. Declare 2 loop variables a and b.
4. Write an outer for loop with variable a, and a nested for loop containing the variable b.
5. As long as b<=a, print ‘(“%d”,b)’. Once b>a, the control returns to the outer loop.
6. Execution continues till outer loop condition is false.

CODE:

```
#include <stdio.h>
int main()
{
   int a, b, rows;
   printf("Enter the number of rows: ");
   scanf("%d", &rows);
   for (a = 1; a <= rows; ++a)
   {
      for (b = 1; b <= a; ++b)
      {
         printf("%d ", b);
      }
      printf("\n");
   }
   return 0;
}
```

OUTPUT:

Enter the number of rows: 9

1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
1 2 3 4 5 6
1 2 3 4 5 6 7
1 2 3 4 5 6 7 8
1 2 3 4 5 6 7 8 9

Program 3: Half pyramid of capital alphabets

ALGORITHM:

1. Accept an input capital character from user.
2. Store the value in a variable (rows).
3. Declare 2 loop variables a and b. Declare a character variable ‘n’.
4. Write an outer for loop with variable a, and a nested for loop containing the variable b.
5. As long as b<=a, print alphabets. Once b>a, the value of ‘n’ is incremented and the control returns to the outer loop.
6. Execution continues till outer loop condition is false.

CODE:

```
#include <stdio.h>
int main()
{
   int a, b;
   char c, m='A';
   printf("Enter a capital letter: ");
   scanf("%c", &c);


for (a = 1; a<=(c-'A'+1); ++a)
```

```
  {
    for (b = 1; b <= a; ++b)
    {
      printf("%c ",m);
    }
    printf("\n");
    m++;
  }
  return 0;
}
```

OUTPUT:

```
Enter a capital letter: F
A
B B
C C C
D D D D
E E E E E
F F F F F F
```

*Program 4: Inverted half pyramid of **

ALGORITHM:

1. Accept an input value in for number of rows from user.
2. Store the value in a variable (rows).
3. Declare 2 loop variables a and b.
4. Write an outer for loop with variable a, and a nested for loop containing the variable b.
5. As long as b<=a, print '*'.
6. Once b>a, the control returns to the outer loop.
7. Execution continues till outer loop condition is false.

CODE:

```
#include <stdio.h>
int main()
{
  int a, b, rows;
  printf("Enter the number of rows: ");
  scanf("%d", &rows);
```

```
for (a = rows; a >= 1; --a)
  {
    for (b = 1; b <= a; ++b)
    {
      printf("* ");
    }
    printf("\n");
  }
  return 0;
}
```

OUTPUT:

```
Enter the number of rows: 9
* * * * * * * * *
* * * * * * * *
* * * * * * *
* * * * * *
* * * * *
* * * *
* * *
* *
*
```

Program 5: Inverted half pyramid of numbers

ALGORITHM:

1. Accept an input value for number of rows from user.
2. Store the value in a variable (rows).
3. Declare 2 loop variables a and b.
4. Write an outer for loop with variable a, and a nested for loop containing the variable b.
5. As long as b<=a, print ‘(“%d”,b)’. Once b>a, the control returns to the outer loop.
6. Execution continues till outer loop condition is false.

CODE:

```
#include <stdio.h>
int main()
{
   int a, b, rows;
   printf("Enter the number of rows: ");
   scanf("%d", &rows);
   for (a = rows; a >= 1; --a)
   {
      for (b = 1; b <= a; ++b)
      {
         printf("%d ", b);
      }
      printf("\n");
   }
   return 0;
}
```

OUTPUT:

```
Enter the number of rows: 9

1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8
1 2 3 4 5 6 7
1 2 3 4 5 6
1 2 3 4 5
1 2 3 4
1 2 3
1 2
1
```

Program 6: Floyd's triangle

ALGORITHM:

1. Accept an input value for number of rows from user.
2. Store the value in a variable (rows).
3. Declare 2 loop variables a and b.

4. Write an outer for loop with variable a, and a nested for loop containing the variable b.
5. As long as b<=a, print ‘(“%d”,b)’. Once b>a, the control returns to the outer loop.
6. Execution continues till outer loop condition is false.

CODE:

```
#include <stdio.h>
int main() {
   int num, a, b, rows;
   printf("Enter the number of rows: ");
   scanf("%d", &rows);

   for (a = 1; a <= rows; a++)
   {
      for (b =1; b <= a; ++b)
      {
         printf("%d ", num);
         ++num;
      }
      printf("\n");
   }
   return 0;
}
```

OUTPUT:

```
Enter the number of rows: 9
0
1 2
3 4 5
6 7 8 9
10 11 12 13 14
15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31 32 33 34 35
36 37 38 39 40 41 42 43 44
```