

- Write a program that prints the following output on the screen [7.5 marks]

```

1  2  3  4  5  6
1  3  5  7  9  11
1  4  7  10 13 16
1  5  9  13 17 21
1  6  11 16 21 26
1  7  13 19 25 31

```

```

int i, j, k;
for (i = 1; i <= 6; i++)
{
    k = 1;
    cout<< k;
    for (j = 1; j < 6; j++)
    {
        k = k + i;
        cout<< k;
    }
    cout<<endl;
}

```

- Write a program that prints the following output on the screen [7.5 marks]

```

1
2  2
1  2  3
4  4  4  4
1  2  3  4  5
6  6  6  6  6  6
1  2  3  4  5  6  7
8  8  8  8  8  8  8  8
1  2  3  4  5  6  7  8  9
10 10 10 10 10 10 10 10 10

```

```

int i, j, k;
for (i = 1; i <= 10; i++)
{
    for (j = 1; j <= i; j++)
    {
        if (i % 2 == 0)
            cout<< i;
        else
            cout<< j;
    }
    cout<<endl;
}

```

- Write a program that prints the following output on the screen [10 marks]

```

      *
      9  9
     8  8  8

```

```

              7  7  7  7
            6  6  6  6  6
          *  *  *  *  *
        4  4  4  4  4  4
      3  3  3  3  3  3  3
    2  2  2  2  2  2  2  2
  1  1  1  1  1  1  1  1  1

int i, j, k;
for (i = 1; i <= 10; i++)
{
    for (k = 1; k <= (10 - i); k++)
    {
        cout<<" ";
    }
    for (j = 1; j <= i; j++)
    {
        if ((i == 1) || (i == 6))
            cout<<"*";
        else
            cout<<k;
    }
    cout<<endl;
}

```

Input an integer containing only 0s and 1s (i.e., a "binary" integer) and print its decimal equivalent. Use the modulus and division operators to pick off the "binary" number's digits one at a time from right to left. Much as in the decimal number system, where the rightmost digit has a positional value of 1, the next digit left has a positional value of 10, then 100, then 1000, and so on, in the binary number system the rightmost digit has a positional value of 1, the next digit left has a positional value of 2, then 4, then 8, and so on. Thus the decimal number 234 can be interpreted as $2 * 100 + 3 * 10 + 4 * 1$. The decimal equivalent of binary 1101 is $1 * 1 + 0 * 2 + 1 * 4 + 1 * 8$ or $1 + 0 + 4 + 8$, or 13. [10 marks]

Decimal to binary conversion

```

int num, rem, temp, dec = 0, b = 1;
cout << "Enter the binary number : ";
cin >> num;
temp = num;
while (temp > 0)
{
    rem = temp % 10;
    cout << "rem: " << rem << endl;
    dec = dec + rem * b;
    cout << "dec: " << dec << endl;
    b *= 2;
    cout << "b: " << b << endl;
    temp /= 10;
    cout << "temp: " << temp << endl;
}
cout << "The decimal equivalent of " << num << " is " << dec;

```