

Exempel 2.1

```
1. // Ett första exempel
2. #include <iostream>
3. using namespace std;

4. int main()
5. {
6.     cout << "Mitt första program ";
7.     return 0;
8. }
```

Exempel 2.2

```
1. // Reserverade tecken och utmatning på två rader
2. #include <iostream>
3. using namespace std;

4. int main()
5. {
6.     cout << "Ett \\\-tecken i en textsträng måste ";
7.     cout << "föregås av ett \\\-tecken." << endl;
8.     cout << "Textsträngar omges med \\\-tecknet.";
9.     return 0;
10. }
```

Exempel 3.1

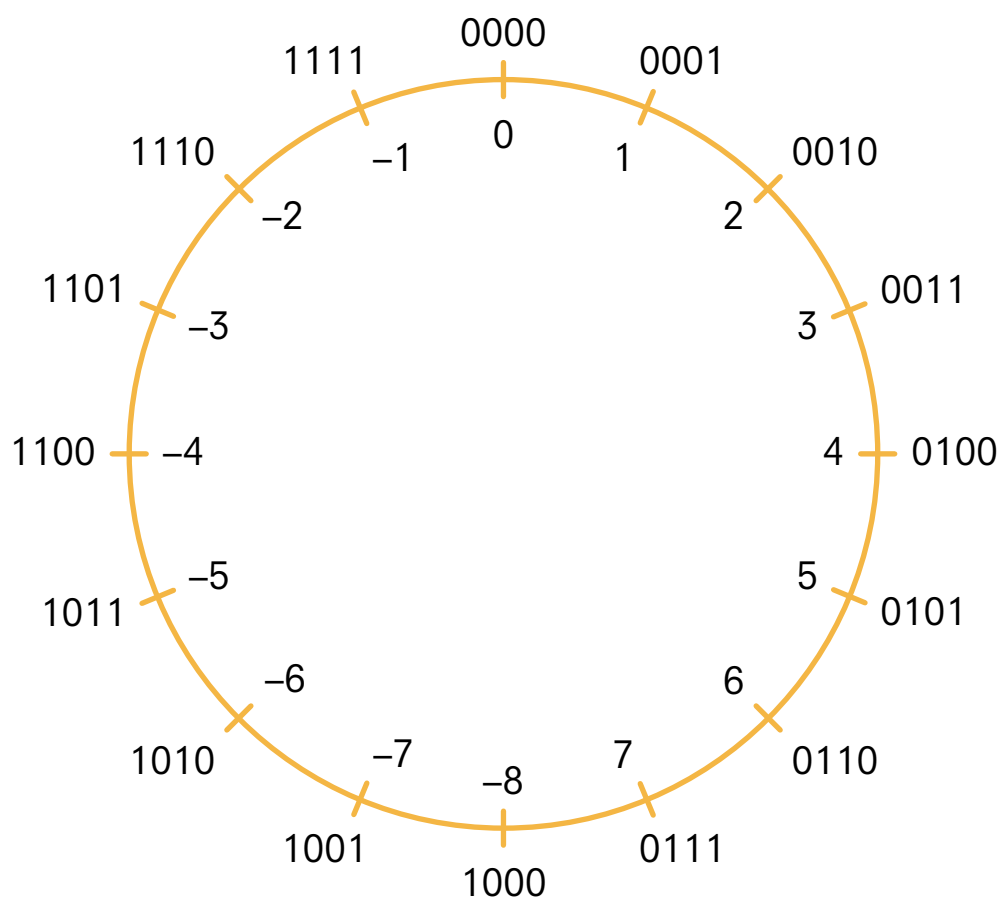
```
1. // tilldelningsoperatörn
2. #include <iostream>
3. using namespace std;

4. int main()
5. {
6.     int a = 8;
7.     int b = 13-a;
8.     b = b-a;
9.     a = a*a;
10.    cout << "a = " << a << ", b = " << b;
11.    return 0;
12. }
```

Exempel 3.2

```
1. // cin
2. #include <iostream>
3. using namespace std;

4. int main()
5. {
6.     int tal1, tal2, tal3;
7.     cout << "Mata in ett tal: ";
8.     cin >> tal1;
9.     cout << "Mata in två tal: ";
10.    cin >> tal2 >> tal3;
11.    cout << "Du har matat in talen " << tal1 << ", "
        << tal2 << " och " << tal3 << endl;
12.    return 0;
13. }
```



Exempel 3.3

```
1. // heltalsvariabler
2. #include <iostream>
3. #include <climits>
4. using namespace std;

5. int main()
6. {
7.     int minnesstorlek = sizeof(int);
8.     int max = INT_MAX;
9.     int min = INT_MIN;
10.    cout << "Antal bytes: " << minnesstorlek << endl;
11.    cout << "Största värde: " << max << endl;
12.    cout << "Minsta värde: " << min << endl;
13.    return 0;
14. }
```

Exempel 3.4

```
1. // char
2. #include <iostream>
3. using namespace std;

4. int main()
5. {
6.     char tecken = 'a';
7.     cout << "tecken = " << tecken << endl;
8.     cout << "Mata in ett nytt tecken: ";
9.     cin >> tecken;
10.    cout << "tecken = " << tecken << endl;
11.    tecken = tecken + 1;
12.    cout << "tecken = " << tecken << endl;
13.    return 0;
14. }
```

Exempel 3.5

```
1. // manipulatorer
2. #include <iostream>
3. #include <iomanip>
4. using namespace std;

5. int main()
6. {
7.     double tal = 4.0/3;
8.     cout << "tal: " << tal << endl;
9.     cout << "tal med tre värdesiffror: "
10.         << setprecision(3) << tal;
11.     cout << "tal med tre decimaler: "
12.         << fixed << setprecision(3) << tal;
13.     return 0;
14. }
```

Exempel 3.6

```
1. // typomvandlingar
2. #include <iostream>
3. using namespace std;

4. int main()
5. {
6.     int tal1;
7.     cout << "Mata in ett heltal: ";
8.     cin >> tal1;
9.     double tal2;
10.    cout << "Mata in ett decimaltal: ";
11.    cin >> tal2;
12.    char tecken;
13.    cout << "Mata in en bokstav: ";
14.    cin >> tecken;
15.    cout << "Vanlig division: " << tal1/tal2 << endl;
16.    cout << "Typomvandlad division: "
17.        << tal1/(int)tal2 << endl;
18.    cout << tecken << " har ASCII-koden "
19.        << (int)tecken;
20.    return 0;
21. }
```

Exempel 3.7

```
1. // fält
2. #include <iostream>
3. using namespace std;

4. int main()
5. {
6.     int vikt[] = {67, 54};
7.     cout << "Innehåll i fältet vikt: "
           << vikt[0] << ' ' << vikt[1] << endl;

8.     int temp = vikt[0];
9.     vikt[0] = vikt[1];
10.    vikt[1] = temp;

11.    cout << "Innehåll i fältet vikt: "
          << vikt[0] << ' ' << vikt[1];
12.    return 0;
13. }
```

The diagram illustrates the execution of the swap function using a temporary variable. It shows three states of the array `vikt` and the variable `temp`:

- Initial state:** `vikt[0]` contains 67, `vikt[1]` contains 54, and `temp` is initialized to 67.
- After `vikt[0] = vikt[1];`:** `vikt[0]` is updated to 54, `vikt[1]` remains 54, and `temp` remains 67.
- After `vikt[1] = temp;`:** `vikt[0]` is 54, `vikt[1]` is updated to 67 (the value stored in `temp`), and `temp` remains 67.

Exempel 3.8

```
1. // strängar
2. #include <iostream>
3. using namespace std;

4. int main()
5. {
6.     cout << "Hej, vad heter du? (ett namn tack): ";
7.     char namn[20];
8.     cin >> namn;
9.     cout << "Ge ditt telefonnummer utan mellanslag: ";
10.    char tel[15];
11.    cin >> tel;
12.    cout << "Hej, " << namn
          << ". Ditt telefonnummer är " << tel;
13.    return 0;
14. }
```

Exempel 4.1

```
1.  // aritmetiska operatorer
2.  #include <iostream>
3.  using namespace std;

4.  int main()
5.  {
6.      int tal1, tal2;
7.      cout << "Mata in två heltal: ";
8.      cin >> tal1 >> tal2;
9.      cout << tal1 << '/' << tal2 << " = "
          << tal1/tal2 << endl;
10.     cout << tal1 << '%' << tal2 << " = "
          << tal1%tal2 << endl;
11.     return 0;
12. }
```

Exempel 4.2

```
1.  // modulusdivision
2.  #include <iostream>
3.  using namespace std;

4.  int main()
5.  {
6.      int tid = 800;
7.      int min = tid/60;
8.      int sek = tid%60;
9.      cout << tid << " s = " << min << " min och "
          << sek << " s.";
10.     return 0;
11. }
```

Exempel 4.3

```
1. // tilldelande aritmetisk operator
2. #include <iostream>
3. using namespace std;

4. int main()
5. {
6.     int produkt = 4;
7.     int tal = 5;
8.     produkt *= tal;
9.     cout << "produkt = " << produkt;
10.    return 0;
11. }
```

Exempel 4.4

```
1. // tillväxtoperatorn
2. #include <iostream>
3. using namespace std;

4. int main()
5. {
6.     int c = 4;
7.     int d = c++;
8.     // d tilldelas värdet 4, därefter får c värdet 5
9.     cout << "c = " << c << ", d = " << d << endl;

10.    int e = ++c;
11.    // c får värdet 6, därefter tilldelas e värdet 6
12.    cout << "c = " << c << ", e = " << e << endl;
13.    return 0;
14. }
```

Exempel 4.5

```
1.  // prioriteringsregler
2.  #include <iostream>
3.  using namespace std;

4.  int main()
5.  {
6.      int tal1 = 4+5;
7.      int tal2 = 2+3*6;
8.      cout << "tal1 = " << tal1 << endl;
9.      cout << "tal2 = " << tal2 << endl;

10.     boolean svar1 = 2+3*2 < 7;
11.     boolean svar2 = 8 > 7 && 9 > 8;
12.     cout << "svar1 = " << svar1 << endl;
13.     cout << "svar2 = " << svar2 << endl;
14.     return 0;
15. }
```
