```
const c =
'EEWHLODBCATECKUFTLWWQEHCUEEWQZUFLQUUZOUHNMYPNFZIILOUHFRPNTGUURCGYUSCEOBUZFLODITAJGYPKVRWJISUWJYFNXGUU
IYUVGOLVPIRYGYTTYWUEDLKAUEXKCJSCQUBUMWTBNXGYPSNHYPNNFPFVYTSVSNGGVDXERFNMKJIELPZJEXKLMEVGUTMCVAFLPGYTOL
IBOGXGYPSNHYPNNOPUTYNZEELBDFINGUEIPKZJOHFPFVYTDVNXFAFNQGYEEHUVGOLVBPNXGYGRIPAZBAGGPGYPZJEMQSMEHKUFULGT
IAORARUUTAJELXLSSITNUUHFIFHUPKFLNYLSDYPBNEOTLTTYNSVNABBTIWJLSNMEOJCEGUXILGADHXKLBCBVLEIPKZJOHFLSSOGKGR
IPABLMXLSSNCLSKOPNCEXGULEHUPFDUUKFRZCSMDYTVTTZTVPTATVTSYHVMGYPMVELFHTLUPKIAVGUXILFKJEMFHSFHKJITAGZDHYJ
LODCGOBUJVZURYKANAWJAEEMHLJNXGZCEZKUEENUPDHHWYOOWJCJELUHHEMOHFRMEOFEHVMFRHVKJETGPUECNA';

const createFrequencyAnalysis = (cypherText, keyLength) => {
  const frequenciesObj = [];
  for (let i = 0; i < keyLength; i++) {
    frequenciesObj[i] = {};
    for (let j = 0; j < Math.floor(cypherText.length / keyLength); j++) {
      const char = cypherText[i + keyLength * j];
      frequenciesObj[i][char] = frequenciesObj[i][char] ? frequenciesObj[i][char] + 1 : 1;
    }
  }
  const frequencies = [];
  frequenciesObj.forEach((frequency) => {
    frequencies.push(Object.keys(frequency)
      .map((key) => [key, frequency[key]])
      .sort((a, b) => b[1] - a[1])
      .map((e) => {
        e.push((e[1] / Object.values(frequency).reduce((acc, cur) => acc + cur, 0)) * 100);
        return e;
      }));
  });
  const maxCharAmount = frequencies.reduce((acc, cur) => Math.max(acc, cur.length), 0);
  frequencies.forEach((e) => {
    while (e.length < maxCharAmount) {
      e.push(['', 0, 0]);
    }
  });
  const masterFrequencies = [];
  for (let i = 0; i < maxCharAmount; i++) {
    masterFrequencies.push(
      frequencies
        .map((e) => e[i][2])
        .reduce((acc, cur) => acc + cur) / frequencies.length,
    );
  }

  console.log('Schlüssel der Länge', keyLength);
  console.log(masterFrequencies);
  console.log(frequencies);
};

// Grossbuchstaben: ASCII 65-90
const decrypt = (cypherText, key) => {
  let decrypted = '';
  let keyPosition = 0;
  for (let i = 0; i < cypherText.length; i++) {
    // console.log();
    // console.log(cypherText[i], cypherText.charCodeAt(i));
    // console.log(key[keyPosition], key.charCodeAt(keyPosition) - 65);
    let nextCode = cypherText.charCodeAt(i) - (key.charCodeAt(keyPosition) - 65);
    // console.log(nextCode);
    if (nextCode > 90) {
      nextCode -= 26;
    }
    if (nextCode < 65) {
      nextCode += 26;
    }
    // console.log(nextCode);
    // console.log(nextCode);
    // console.log(String.fromCharCode(nextCode));
    decrypted += String.fromCharCode(nextCode);
    keyPosition = (keyPosition + 1) % key.length;
  }
  console.log(decrypted);
};

// for (let i = 1; i <= 10; i++) {
//   createFrequencyAnalysis(c, i);
// }

// createFrequencyAnalysis(c, 5);

decrypt(c, 'BAUCH');

/*
Für verschiedene Schlüssel-Längen die Wahrscheinlichkeit aller Buchstaben des n-ten Zeichen des
Schlüssels nehmen. Dann jeweils der Durschnitt der  n-häufigsten Wahrscheinlichkeiten nehmen,
sortieren und ausgeben:

Schlüssel der Länge 1
[
   7.226890756302522,  6.722689075630252,
   5.714285714285714,  5.546218487394958,
   5.042016806722689,  4.873949579831933,
   4.873949579831933,  4.873949579831933,
   4.705882352941177,   4.53781512605042,
   3.865546218487395,  3.865546218487395,
  3.5294117647058822, 3.5294117647058822,
  3.5294117647058822,  3.361344537815126,
   3.361344537815126,  2.857142857142857,
   2.857142857142857,  2.689075630252101,
  2.5210084033613445, 2.5210084033613445,
  2.1848739495798317, 2.1848739495798317,
  2.0168067226890756, 1.0084033613445378
]
Schlüssel der Länge 2
[
   7.239057239057239, 6.7340067340067336,
   5.892255892255893,  5.892255892255893,
   5.723905723905724,  5.218855218855218,
    5.05050505050505,   5.05050505050505,
   4.377104377104377,  4.208754208754209,
   4.040404040404041,  3.872053872053872,
  3.7037037037037033, 3.7037037037037033,
  3.3670033670033668, 3.3670033670033668,
  3.0303030303030303, 2.6936026936026933,
  2.6936026936026933,  2.525252525252525,
   2.356902356902357, 2.1885521885521886,
  2.0202020202020203, 2.0202020202020203,
  2.0202020202020203,   1.01010101010101
]
Schlüssel der Länge 3
[
   8.417508417508417,  6.228956228956229,
   5.892255892255892,  5.892255892255892,
   5.387205387205387,  5.387205387205387,
   5.050505050505051,  4.882154882154882,
   4.377104377104377,  4.377104377104377,
  4.2087542087542085, 4.2087542087542085,
   3.872053872053872, 3.7037037037037037,
   3.535353535353535, 3.1986531986531985,
    3.0303030303030303, 2.8619528619528616,
   2.356902356902357,  2.356902356902357,
   2.356902356902357, 2.0202020202020203,
  2.0202020202020203, 2.0202020202020203,
  1.3468013468013467, 1.0101010101010102
]
Schlüssel der Länge 4
[
   8.108108108108109,  7.094594594594595,
   6.587837837837838,               6.25,
   5.743243243243244,  5.405405405405405,
  5.0675675675675675, 4.5608108108108105,
  4.5608108108108105, 4.5608108108108105,
   4.391891891891892,  4.054054054054054,
   3.547297297297298, 3.3783783783783785,
  3.3783783783783785, 3.2094594594594597,
  3.0405405405405403,  2.871621621621622,
  2.7027027027027026, 2.5337837837837838,
   2.364864864864865,  2.027027027027027,
  1.8581081081081081, 1.3513513513513513,
  0.8445945945945945, 0.5067567567567568
]
Schlüssel der Länge 5
[
  16.302521008403364,  10.756302521008404,
   9.243697478991598,   8.403361344537815,
   7.394957983193278,   6.554621848739496,
   6.050420168067227,   5.378151260504202,
    4.53781512605042,  3.8655462184873954,
  3.5294117647058827,   3.361344537815126,
   3.025210084033614,  2.1848739495798317,
  1.8487394957983194,   1.680672268907563,
   1.680672268907563,  1.3445378151260505,
  1.0084033613445378,  1.0084033613445378,
  0.5042016806722689, 0.33613445378151263
]
Schlüssel der Länge 6
[
   8.754208754208754,  7.575757575757577,
   6.902356902356902,  6.060606060606061,
   5.555555555555556, 5.3872053872053876,
   4.882154882154882,  4.882154882154882,
   4.545454545454546, 4.2087542087542085,
   4.040404040404041, 3.8720538720538724,
  3.8720538720538724,  3.535353535353536,
   3.367003367003367,  3.198653198653199,
   3.198653198653199, 2.6936026936026938,
   2.356902356902357,  2.356902356902357,
  2.0202020202020203,  1.851851851851852,
  1.6835016835016836, 1.5151515151515154,
  1.0101010101010102, 0.6734006734006734
]
Schlüssel der Länge 7
[
     9.91596638655462,   7.731092436974791,
    6.890756302521008,   6.386554621848739,
    6.218487394957983,   5.546218487394958,
   5.2100840336134455,   4.705882352941177,
    4.705882352941177,   4.537815126050421,
    4.201680672268908,   4.033613445378151,
   3.8655462184873945,   3.529411764705882,
    3.361344537815126,   3.025210084033614,
    2.857142857142857,   2.521008403361345,
    2.184873949579832,  1.8487394957983196,
   1.8487394957983196,  1.8487394957983196,
   1.3445378151260503,  1.1764705882352942,
  0.33613445378151263, 0.16806722689075632
]
Schlüssel der Länge 8
[
   9.966216216216218,   8.277027027027028,
   7.263513513513514,   6.587837837837838,
   6.081081081081081,   5.574324324324323,
   5.405405405405404,  5.0675675675675675,
   4.898648648648648,    4.72972972972973,
   4.054054054054055,   3.885135135135136,
   3.885135135135136,   3.378378378378378,
   2.871621621621621,   2.702702702702702,
  2.5337837837837833,  2.5337837837837833,
  2.3648648648648645,  2.0270270270270268,
   1.689189189189189,  1.3513513513513513,
  1.1824324324324322,  1.0135135135135134,
  0.5067567567567568, 0.16891891891891891
]
Schlüssel der Länge 9
[
   10.26936026936027,   8.417508417508419,
   7.744107744107745,   6.565656565656567,
   6.060606060606061,   5.892255892255893,
   5.892255892255893,   5.050505050505051,
  4.7138047138047146,   4.545454545454546,
   4.377104377104377,   3.703703703703704,
   3.198653198653199,  3.0303030303030307,
  3.0303030303030307,  2.8619528619528625,
  2.8619528619528625,  2.1885521885521886,
   1.851851851851852,  1.5151515151515154,
  1.5151515151515154,  1.5151515151515154,
  1.3468013468013469,  1.0101010101010102,
  0.6734006734006734, 0.16835016835016836
]
Schlüssel der Länge 10
[
  16.779661016949152,  11.35593220338983,
  10.169491525423728,   8.47457627118644,
   7.288135593220336,  6.949152542372879,
   5.762711864406779,  5.254237288135593,
   4.745762711864407,  4.237288135593221,
  3.8983050847457625, 3.2203389830508464,
    2.71186440677966, 2.5423728813559316,
  2.0338983050847452, 1.6949152542372876,
    1.35593220338983, 1.0169491525423726,
  0.3389830508474576, 0.1694915254237288
]

=> Schlüssel der Länge 5.

Wahrscheinlichkeiten der Buchstaben bei Schlüssellänge 5
(der Buchstaben, die jeweils mit dem n-ten Buchstaben des Schlüssels verschlüsselt werden.):

Format: [Zeichen im Cyphertext, Anzahl Vorkommnisse, Anteil am Text]
```

```
[
  [
    [ 'F', 17, 14.285714285714285 ],
    [ 'J', 12, 10.084033613445378 ],
    [ 'U', 10, 8.403361344537815 ],
    [ 'E', 9, 7.563025210084033 ],
    [ 'T', 9, 7.563025210084033 ],
    [ 'P', 8, 6.722689075630252 ],
    [ 'S', 8, 6.722689075630252 ],
    [ 'O', 6, 5.042016806722689 ],
    [ 'B', 6, 5.042016806722689 ],
    [ 'V', 6, 5.042016806722689 ],
    [ 'I', 5, 4.201680672268908 ],
    [ 'G', 4, 3.361344537815126 ],
    [ 'M', 4, 3.361344537815126 ],
    [ 'N', 3, 2.5210084033613445 ],
    [ 'D', 3, 2.5210084033613445 ],
    [ 'X', 2, 1.680672268907563 ],
    [ 'C', 2, 1.680672268907563 ],
    [ 'Q', 1, 0.8403361344537815 ],
    [ 'Z', 1, 0.8403361344537815 ],
    [ 'R', 1, 0.8403361344537815 ],
    [ 'L', 1, 0.8403361344537815 ],
    [ 'H', 1, 0.8403361344537815 ]
  ],
  [
    [ 'E', 26, 21.84873949579832 ],
    [ 'N', 12, 10.084033613445378 ],
    [ 'R', 9, 7.563025210084033 ],
    [ 'S', 9, 7.563025210084033 ],
    [ 'D', 8, 6.722689075630252 ],
    [ 'U', 8, 6.722689075630252 ],
    [ 'T', 7, 5.88235294117647 ],
    [ 'I', 7, 5.88235294117647 ],
    [ 'O', 6, 5.042016806722689 ],
    [ 'G', 4, 3.361344537815126 ],
    [ 'L', 4, 3.361344537815126 ],
    [ 'H', 4, 3.361344537815126 ],
    [ 'A', 3, 2.5210084033613445 ],
    [ 'F', 2, 1.680672268907563 ],
    [ 'Z', 2, 1.680672268907563 ],
    [ 'M', 2, 1.680672268907563 ],
    [ 'V', 2, 1.680672268907563 ],
    [ 'C', 2, 1.680672268907563 ],
    [ 'B', 1, 0.8403361344537815 ],
    [ 'K', 1, 0.8403361344537815 ],
    [ '', 0, 0 ],
    [ '', 0, 0 ]
  ],
  [
    [ 'Y', 16, 13.445378151260504 ],
    [ 'L', 14, 11.76470588235294 ],
    [ 'H', 12, 10.084033613445378 ],
    [ 'X', 11, 9.243697478991598 ],
    [ 'M', 9, 7.563025210084033 ],
    [ 'N', 9, 7.563025210084033 ],
    [ 'W', 7, 5.88235294117647 ],
    [ 'C', 7, 5.88235294117647 ],
    [ 'O', 5, 4.201680672268908 ],
    [ 'I', 5, 4.201680672268908 ],
    [ 'Z', 4, 3.361344537815126 ],
    [ 'A', 4, 3.361344537815126 ],
    [ 'U', 4, 3.361344537815126 ],
    [ 'P', 3, 2.5210084033613445 ],
    [ 'B', 2, 1.680672268907563 ],
    [ 'T', 2, 1.680672268907563 ],
    [ 'V', 2, 1.680672268907563 ],
    [ 'Q', 1, 0.8403361344537815 ],
    [ 'E', 1, 0.8403361344537815 ],
    [ 'J', 1, 0.8403361344537815 ],
    [ '', 0, 0 ],
    [ '', 0, 0 ]
  ],
  [
    [ 'G', 23, 19.327731092436977 ],
    [ 'K', 12, 10.084033613445378 ],
    [ 'P', 11, 9.243697478991598 ],
    [ 'F', 10, 8.403361344537815 ],
    [ 'T', 9, 7.563025210084033 ],
    [ 'U', 7, 5.88235294117647 ],
    [ 'V', 7, 5.88235294117647 ],
    [ 'J', 6, 5.042016806722689 ],
    [ 'H', 5, 4.201680672268908 ],
    [ 'C', 4, 3.361344537815126 ],
    [ 'W', 4, 3.361344537815126 ],
    [ 'Q', 4, 3.361344537815126 ],
    [ 'E', 4, 3.361344537815126 ],
    [ 'N', 3, 2.5210084033613445 ],
    [ 'I', 2, 1.680672268907563 ],
    [ 'O', 2, 1.680672268907563 ],
    [ 'B', 2, 1.680672268907563 ],
    [ 'X', 2, 1.680672268907563 ],
    [ 'R', 1, 0.8403361344537815 ],
    [ 'Y', 1, 0.8403361344537815 ],
    [ '', 0, 0 ],
    [ '', 0, 0 ]
  ],
  [
    [ 'L', 15, 12.605042016806722 ],
    [ 'U', 14, 11.76470588235294 ],
    [ 'A', 13, 10.92436974789916 ],
    [ 'Y', 11, 9.243697478991598 ],
    [ 'Z', 9, 7.563025210084033 ],
    [ 'K', 7, 5.88235294117647 ],
    [ 'P', 7, 5.88235294117647 ],
    [ 'V', 6, 5.042016806722689 ],
    [ 'H', 5, 4.201680672268908 ],
    [ 'O', 4, 3.361344537815126 ],
    [ 'S', 4, 3.361344537815126 ],
    [ 'B', 4, 3.361344537815126 ],
    [ 'N', 3, 2.5210084033613445 ],
    [ 'R', 2, 1.680672268907563 ],
    [ 'I', 2, 1.680672268907563 ],
    [ 'C', 2, 1.680672268907563 ],
    [ 'T', 2, 1.680672268907563 ],
    [ 'G', 2, 1.680672268907563 ],
    [ 'J', 2, 1.680672268907563 ],
    [ 'D', 2, 1.680672268907563 ],
    [ 'M', 2, 1.680672268907563 ],
    [ 'W', 1, 0.8403361344537815 ]
  ]
]

=> Verschiebungen um [1, 0, 20, 2, 7]
=> Schlüssel: BAUCH.

=> Klartext:
DECFENDHATSEIINETRUPPENANDECOSTFRONTZUSAMMENGEZOGENUNDKONZENTRIERTSICHAUFDENDORTIGENDURCHBRUCHREN
DENTIESOFORTIHREERSTEUNDDRITTEDIVISIONAUSUMANDEROSTFRONTDIEVERLUSTEZUDDCKENSICHERNSIEDIE
LEBENSMITTELVERSORGUNGDEROSTFRONTMITTELSDERZWEITENDIVISIONDIEVERWUNDDTENWERDENSOFORTUONDERFRONT
YBGEZOGENSIESOLLENINEUREMHAUPTQUARTIERVERSORGTUNDBEHANDELTWERDENUMEURESTELLUNGZUSICHERNSCHICKEN
WIRETCHDIEACHTEDIVISIONDERSUEDFRONTALSVERSTAERKUNGBEDENKENSIEDASDERFALLDEROSTFROOTGROSSEFOLGEN
FUERDASLANDHABENWIRDDIESDARFNICHTGESCHEHENDIEHAUPTSTREITMACHTDESFEINDESBEFINDETSICHNURNOCH
VIERSAGESMAERSCHEENTFERNTDIEZEITEILT
*/
```