

```

diferir
retornar b [1] - a [1];// classificar por maior
contar.
}
});
// converte as contagens em porcentagens
para (deixe a entrada de entradas) {
  entrada [1] = entrada [1] / this.Totalletters*100;
}
// soltar qualquer entradas inferiores a 1%
entradas = entradas.Filter (Entrada => Entrada [1]> = 1);
// agora converte cada entrada em uma linha de texto
Deixe linhas = entradas.map (
  ([l, n]) => `$ {l}: $ {"#". Repita (math.round (n))}`
  $ {n.toFixed (2)}%`
);
// e retorne as linhas concatenadas, separadas por
personagens newline.
retorno linhas.Join ("\ n");
}
}
// Esta função Async (Promise Returning) cria um histograma
objeto,
// lê de forma assíncrona pedaços de texto da entrada padrão e
adiciona esses pedaços a
// o histograma.Quando chega ao final do fluxo, ele
Retorna este histograma
Função assíncrona histogramfromstdin () {
  process.stdin.setEncoding ("UTF-8");// Leia Unicode
  cordas, não bytes
  Seja histograma = novo histograma ();
  para aguardar (let chunk of process.stdin) {
    histogram.add (pedaço);
  }
  histograma de retorno;
}
// Esta linha final de código é o principal corpo do programa.

```