```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <assert.h>

#define NN 1000000
#define eps 1e-7

FILE *fi, *fo;
int n;
double C;
double a[NN], p[NN];
int h[NN];
int s, t, i, j;
double l, r;

/* В h[s]  хранится число k, такое что a[k] является максимальным элементом среди
   a[i], a[i + 1], ... a[j - 1].
   В h[s + 1]  хранится число k, такое что a[k] является максимальным элементом среди
   a[h[s] + 1], a[h[s] + 2], ... a[j - 1].
   И так далее.
   В h[t - 1] всегда хранится j - 1. */

double
get_h (void)
{
  assert (s < t);
  return a[h[s]];
}

int
pop_h (void)
{
  assert (s < t);
  return h[s++];
}

void
push_h (int x)
{
  while (s < t && a[h[t - 1]] < a[x])
    t--;
  h[t++] = x;
}

/* Поставим на отверстия кран и будем их открывать по одному, ждать пока
   уровень воды придет к равновесию, а затем открывать следующий.

   Рассмотрим, что происходит, когда мы открываем следующий кран. Сначала
   систему можно рассматривать как два прямоугольных сообщающихся сосуда. Это
   свойство сохраняется до тех пор, пока:
     1) уровень воды в левом не опустится до уровня одного из кранов,
```

находящихся в левом сосуде;
2) уровень воды в левом и правом не выравнятся.
В первом случае было достигнуто состояние равновесия и нужно переходить к следующему крану. Во втором нужно изменить конфигурацию левого сосуда. */

```
int
main (int argc, char *argv[])
{
  double ev, ev1, ev2;
  int x;
  fi = fopen ("d.in", "rt");
  assert (fi != NULL);
  fo = fopen ("d.out", "wt");
  assert (fo != NULL);

  fscanf (fi, "%d%lf", &n, &C);
  for (i = 0; i < n - 1; i++)
    fscanf (fi, "%lf", &a[i]);

  for (i = 0; i < n; i++)
    p[i] = 0;

  /* Левый сосуд - отделения с i по j - 1, правый - отделение j.
     Уровень жидкости в левом - l,  в правом - r. */
  s = 0; t = 0;
  i = 0; j = 1;
  l = C; r = 0;
  push_h (0);

  while (j < n && i < j && l > a[j - 1] + eps)
    {
      /* Подсчитаем, сколько воды должно перетечь из левого сосуда в правый
     для наступления событий 1 и 2. */
      ev1 = (l - r) * (j - i) / (j - i + 1);
      if (s < t)
    {
      ev2 = (l - get_h ()) * (j - i);
      ev = (ev1 < ev2) ? ev1 : ev2;
    }
      else
    ev = ev1;
      /* Изменим уровень воды в соответствии с ближайшим событием. */
      l -= ev / (j - i);
      r += ev;
      assert (l + eps > r);
      /* Если уровни воды выровнялись, расширим левый сосуд и откроем
     следующий кран. */
      if (fabs (l - r) < eps)
    {
      push_h (j);
      j++;
      r = 0;
```

```
    }
     /* Если уровень воды опустился до самого верхнего крана, расположенного
     внутри левого сосуда, изменим конфигурацию левого сосуда и заполним часть
     ответа.  */
     if (fabs (l - get_h ()) < eps)
    {
     x = pop_h ();
     for (; i <= x; i++)
       p[i] = l;
     i = x + 1;
    }
     /* Если левый сосуд коллапсировал до нулевого объема, значит было
    достигнуто состояние равновесия, когда уровень воды в левом сосуде
    опустился до уровня отверстия, соединяющего его с правым. В этом
    случае правый сосуд становится новым левым, а мы открываем следующий
    кран. */
     if ((i == j) || (l < a[j - 1] + eps))
    {
     for (; i < j; i++)
       p[i] = l;
     l = r;
     r = 0;
     i = j;
     push_h (j);
     j++;
    }
    }
  for (; i < j; i++)
    p[i] = l;

  for (i = 0; i < n; i++)
    fprintf (fo, "%0.20lf\n", p[i]);

  fclose (fi);
  fclose (fo);
  return 0;
}
```