

6925 Volume of Revolution

Revolving a profile curve around an axis can produce many real life objects. The chess pawn piece and the wine glass in the following figures are two such examples:

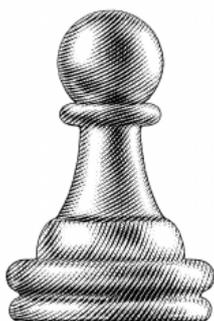


Figure 1a) Chess pawn

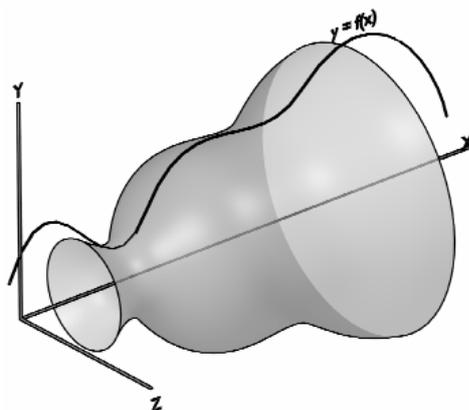


Figure 1c) Generating a 3D object by revolving a profile curve $y = f(x)$ around the x -axis



Figure 1b) Wine glass

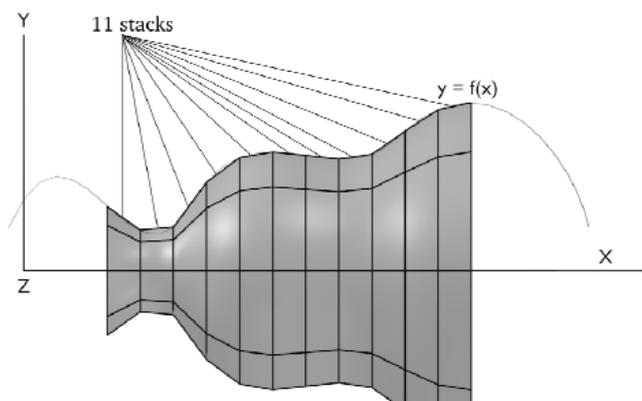


Figure 1d) Side view of a polygon mesh generated by revolving $y = f(x)$ using 8 slices and 11 stacks

To render such objects in Computer Graphics applications (games, animations, etc.) we build polygon meshes to approximate the geometry of the object. One way to build such a mesh is to divide the object into a number of slices (segments around the x axis) and a number of stacks (segments along the x axis). While low polygon count for the mesh results in faster render time, they lack in fidelity. Figure 1d and 2b shows the effect of varying the number of stacks and slices to generate such polygon meshes:

These polygon meshes are often good enough to pass the human eye for a real object as we would employ other faking techniques like per vertex normal computation, texture mapping, etc. However, none of those are relevant for this problem. Here we are only interested to know how “good” a polygon

mesh is in terms of volume encapsulated when we revolve a profile curve within a given range. **Our metric of “goodness” is the volume of the revolution.**

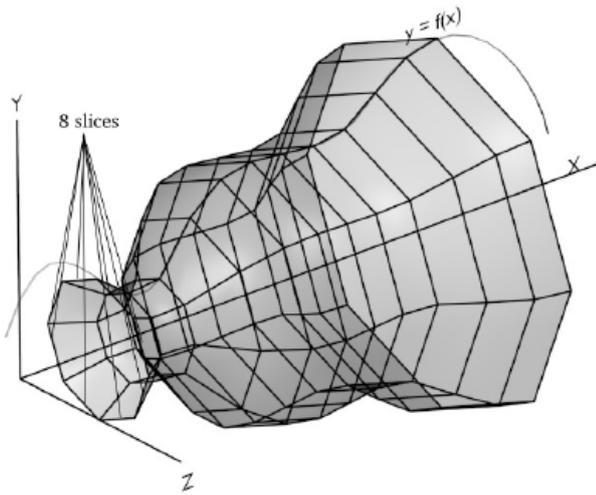


Figure 2a) Same polygon mesh viewed from a different angle

The actual volume of the revolution can be found by using a simple integration formula:

$$V = \int_a^b A(x)dx$$

where $A(x)$ is the area of the circle that we get by taking a cross section of the solid at a given x value. This formula gives us the volume of the object bounded by $x = a$ and $x = b$ planes.

Now the volume of the approximated polygon mesh can be computed too if we are told how many slices and how many stacks we want to divide the solid into. For the sake of simplicity, in this problem we will use polynomials as profile curves. Our polygon mesh will have equally spaced stacks and equal slices. The output you need to print is the true relative error as a percentage found by the formula

$$\left| \frac{\text{TrueValue} - \text{ApproximateValue}}{\text{TrueValue}} \right| \times 100.$$

Input

The input description for the problem will start with T ($1 < T \leq 10000$) — the number of test cases. Each of the following T lines of the input will describe a test case.. The first number in each line will give you the value of n ($0 < n < 6$) — the degree of the polynomial. Then the next $n + 1$ values (separated by white spaces) will give you the coefficients. The coefficients will be listed in the decreasing order of power and each of these coefficients will be in the range $[-10, +10]$. Following the polynomial description you’ll have two numbers giving the value of $x = a$ and $x = b$ ($-10 \leq a < b \leq +10$) on the same line. The last two numbers of each line will give you the number of slices ($3 \leq \text{slices} \leq 100$) and number of stacks ($0 < \text{stacks} \leq 100$) to use in the polygon mesh. You can assume that the curve corresponding to the polynomial equation does not cross the x -axis for the range $[a, b]$.

Output

Output for each test case will start with the test case label (starting with 1, and formatted as shown in sample output.) The label will be followed by true relative error in volume computation as a percentage. You need to round off the result to four digits after the decimal point. You can assume that inputs will be such that small precision errors will not cause difference in the output values.

Sample Input

```
2
2 1 -4 5 1 4 4 3
1 1 0 1 4 4 3
```

Sample Output

```
Case 1: 27.9042
Case 2: 36.3380
```