



PI의 근사값과 오차 구하기 와

Delta = 1.000e-006일 때 n*와 pn 출력하기

결과와 코드 작성 과정



과목명: 시뮬레이션 기초 및 실습

교수: 김지범

학과: 컴퓨터공학부

학번: 202201479

이름: 박지원

제출일: 2023년 04월 01일

목차

I. 문제 1번

- i. 결과 스크린 샷
- ii. 코드 작성 과정

II. 문제 2번

- i. 결과 스크린 샷
- ii. 코드 작성 과정

III. 마무리

I. 문제 1번 _ i. 결과 스크린 샷

```
정수 n 입력:10000
```

```
ans =
```

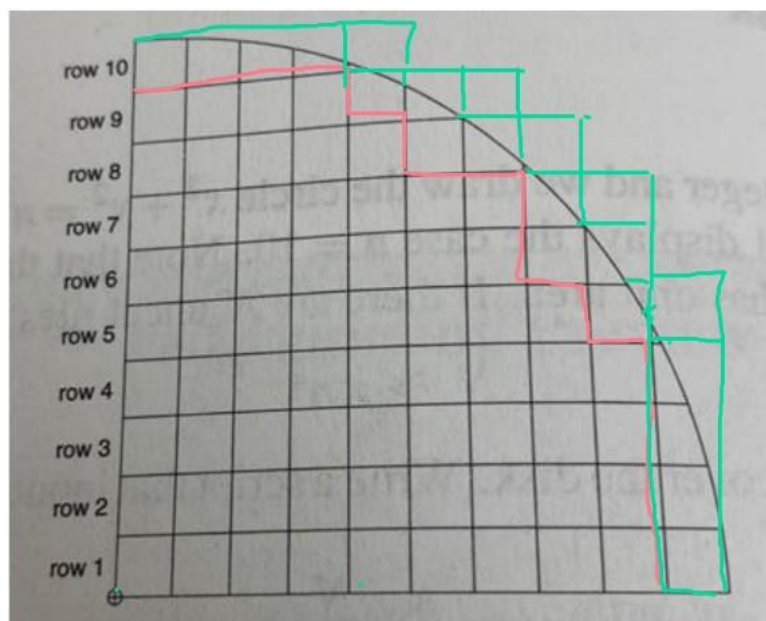
```
19982
```

```
rho_n= 3.14139034
```

```
error= 0.00020231
```

I. 문제 1번 _ ii. 코드 작성 과정

이 코드에서는 cut tile 인 B를 잘 정하는 것이 중요하다. 코드를 작성하기 전에, B를 어떻게 정할 것인지 생각해보았다.



보다 쉽게 이해하기 위해 위의 그림처럼 잘린 타일에 남은 부분을 이어 그려보았다. Cut tile 의 나머지 타일 부분을 이어 그리다 보니, 떠오른 아이디어가 있었다. ceil 함수에 넣은 x 의 값은 잘린 타일까지, 즉 그림처럼 남은 부분 까지를 포함한다. floor 함수에 넣은 x 의 값은 잘리지 않은 타일까지 이다. 따라서 $\text{ceil}(x) - \text{floor}(x)$ 를 하면 위 그림의 파란 타일들을 가리키게 된다. 하나 하나의 타일이 아닌 전체적으로 보면, 대략 파란 부분의 $1/2$ 이 되는 부분이 cut tile 에 해당한다.

따라서 이런 생각을 기반으로 코드를 짰다. Uncut tile 을 $N1$ 이라 하고, $N1$ 을 초기화 한다. Input 을 이용해 n 을 입력 받고, B 를 초기화 한다. for 문을 이용하여 k 가 1 에서 n 까지 일 때까지 반복하는 반복문을 작성한다. 이 반복문에는 x 의 값을 floor 함수에 대입한 un cut tile 을 가리키는 변수 m , $x-m$ 의 값을 ceil 함수에 대입한 cut tile 을 가리키는 B 이때 기존 B 에 $\text{ceil}(x-m)$ 을 증가시키는 식으로 작성한다. 이어서 $N1$ 에 $N1+m$ 을 해준 값을 넣어준다. 반복문을 빠져나오면 최종적인 B 를 2 로 나눠준다. Cut tile 의 개수가 얼마 나오는지 확인하기 위해 $B*4$ 를 작성한다. 이때 B 는 위의 그림처럼 원의 4 분의 1 부분만의 cut tile 이므로 4를 곱해준다. 파이의 근사값, pn 은 $4*(N1+B)/n^2$ 라고 한다. 그리고 $\text{abs}(\pi - pn)$ 을 err 에 넣어 오차를 표현해준다. 마지막으로 fprintf 를 이용하여 파이의 근사값과 오차를 출력한다.

II. 문제 2번 _ i. 결과 스크린 샷

```
enter delta: 1.000e-006
*n = 346.0000000000000000000000
pn = 3.141549490003078
```

II. 문제 2번 _ ii. 코드 작성 과정

양의 임계치 δ 를 input 함수로 입력 받는다. 그리고 최초의 n 은 3으로 초기화한다. 이때 3으로 초기화 하는 이유는, 최소 점이 3개여야지 내접 다각형이 생성될 수 있기 때문이다. 내접 다각형의 넓이를 A_n 이라 하고 $n/2 \cdot \sin(2\pi/n)$ 이라 한다. 외접 다각형의 넓이를 B_n 이라 하고 $n \cdot \tan(\pi/n)$ 이라 한다. Error는 외접 다각형의 넓이에서 내접 다각형의 넓이를 뺀 값이다. 그리고 $n+1$ 을 해준 내접 다각형의 넓이를 A_{n1} 이라 하고 n 에 1을 더해준다. 마찬가지로 B_n 의 식 중 n 에 1을 증가해 준 값을 B_{n1} 이라고 선언해준다. while문을 이용한 반복문을 쓰는데 이때 조건을 $\text{abs}(A_{n1} - A_n) \geq \delta \parallel \text{abs}(B_{n1} - B_n) \geq \delta$ 라고 쓴다. 반복문 속 코드를 살펴보면, 우선 n 의 개수를 증가시켜준다. 그리고 반복문 위에서 작성한 A_n , B_n , A_{n1} , B_{n1} , error를 모두 재정의 해준다. 반복문을 빠져나와 nstar에 n 을 대입해주고, nstar와 pn을 fprintf 함수로 출력해준다.

Ⅲ. 마무리

2번 문제의 경우, 수업시간에 배운 코드 중 while의 조건만 살짝 바꾸면 되는 비교적 쉬운 문제였다. 하지만 1번에서 B를 어떻게 정의할지 정말 오래 고민했다. B만 정의하면 나머지 코드를 작성하는 데는 큰 어려움 없어 보였다. 지난 레포트에서도 cos함수를 직접 그려 범위를 나누었던 시각화를 이번에도 사용하였다. Cut tile의 남은 타일마저 직접 이어서 그려보고, ceil 함수를 이용할 것을 염두 하니, cut tile에 이어진 타일만을 남길 방법이 떠올랐다. 처음부터, 잘린 타일에서 잘리지 않은 타일을 빼면 B가 나오겠다는 방향을 정했던 덕분인지 끈질기게 코드를 작성할 수 있었다. 처음에 코드를 짤 때, for문 속에 B를 초기화 해버려서 자꾸만 B의 값이 0으로 나왔던 실수를 했었다. 이 실수를 알아차렸을 때, 머리속에 엄청나게 각인이 되어 다음에는 이런 실수를 하지 않겠다고 다짐할 수 있었던 좋은 기회였다. 오래 오래 고민하니 코드에 대해 더 잘 이해할 수 있었고, 무엇보다 오랜 고민 끝에 코드를 작성하면 그보다 더 뿌듯할 수 없다. 이렇게 레포트를 작성하며 정리까지 하니 머릿속에도 확실히 각인된다.