COSAT 3rd STEP : CODING BASIC

Due date : 11/30

안녕하세요 선생님! 드디어 코딩 과제를 드리게 되었습니다. 사실 코딩실력은 코딩을 많이해야 빨리 늡니다. 물론 코딩이 컴퓨터 공학도에게 있어서 모든 것은 아니지만 능숙하면 편해지는 부분이 많죠. 이번에 드릴 과제 내용은 사실 그렇게 흥미로운 내용은 아닙니다… 하지만 로직과 메모리의 기본개념을 다지는데 아주 적절하다 생각되어 저희 학교 과제를 발췌해 왔습니다. 두 과제모두 100점 맞은 파일이 있으니 채점의 신뢰성에서는 걱정하지 않으셔도 됩니다.

과제의 설명은 첨부된 pdf파일을 따르시면 되구요, 올바른 코딩습관을 위해서 몇가지 추가로 요구사항이 있습니다.

1. 주석의 중요성 – **거의 모든 라인에 주석을 다세요!**

실제로 이정도로 많이 다는 경우는 많지 않지만, 이번 경우 에는 선생님이 갑작스럽게 어느정도 규모가있는(?) 과제를 하게되었기 때문에 주석을 꼼꼼히 달면서 흐름을 잘파악하시고, 나중에 문제점을 찾을 때 큰 도움이 된다는 점을 직접 부딪혀서 체감 하셨으면 합니다.

1. 코드를 작성하기 전에 계획 짜고, 진행상황 실시간으로 기록하기 – **노트나 파일 제출!**

한번에 막힘없이 코드를 짜고 버그가 하나도 없다면 얼마나 좋을까요? 안타깝지만 그런 경우는 잘 없습니다. 머리가 잘 돌아 간다고 내키는 대로 술술 적다가 어딘가 에서 오류가 생긴다? 그때부터 머리가 아파집니다. 그리고 이러한 과제들 에는 명백한 요구사항들이 있죠? 요구사항들을 잘 따르고 있는지 꼼꼼히 체크해야 정확한 결과물이 나오겠죠. 그래서 제가 추천 드리는 방법은 일단 노트 등에 큰 거부터 작은 거 순으로 어떤 로직으로 작동시킬지, 변수를 어떤 식으로 사용할지, 예외사항이나 특이케이스가 있는지 함수 클래스 등등의 요구사항을 정리합니다. 그리고 코딩을 하면서는 요구사항을 구현 할 때마다 수시로 줄을 긋거나 체크를 합니다. 번거로울거 같지만 그냥저냥 되는대로 짜다가 나중에 버그 찾느라고 통째로 뜯어 고치는거 보다 훨씬 시간이 절약되고 확신을 가지면서 코딩을 할 수 있습니다. (경험담)

1. 테스트 케이스 만들기 – **테스트 코드 포함해서 제출!**

버그를 다 잡았다고 끝나는게 아니지요? 내가 구현한 로직에 빈틈이 있다면 “edge case” 에 발목을 잡히게 됩니다. 경우의 수를 극한으로 따져서 이것도 돌아가나? 이렇게 넣으면 어떻게되지? 끊임없이 검증해야 합니다.

TIP : 코드 다 짜고 한꺼번에 디버깅하기 vs 함수, 클래스 하나 짤때마다 디버깅하기

안 봐도 답이 나오죠? 의외로 실천하기 어렵습니다. 하지만 귀찮은 짓을 미리 많이 할수록 나중에 고통받지 않습니다.

요약 : 뭐뭐 제출해야되나요?

1. PDF파일에 맞게 작성된 코드파일들 (내용을 잘 읽어볼 것. 무엇을 해야하고 하면 안되는지) + 주석
2. 코드 설계&진행상황 자료 (양식과 방법은 본인의 판단으로 추가하거나 바꿔도됨)
3. 테스트 코드 (pdf파일에는 제출안해도 된다고 하는데 그거는 채점과 무관하다는 뜻)

모호한 경우에 대해서 질문은 언제든 환영이며 열심히 따라와 주셔서 감사합니다.