고려대학교
KOREA UNIVERSITY

Object Oriented Programming and Practice, Final Exam

DCCS213(02)

Your ID#: [ ]

Your Name: [ ]

Read each question carefully and do the programming! Be sure to follow the instructions below!

[Instruction] 안내

(1) **Javadoc comment** and **standard naming rule** are essential for all program (**If failure -1 point**).
(Javadoc 코멘트와 스탠다드 네이밍 룰은 이제 말하지 않아도 기본입니다. 실패 시 각 -1점 감점).

(2) Please compress (.zip) and submit this report (.docx), and the entire Java code (.java file) you wrote into the BlackBoard.
(**이 시험지와 답변에 사용된 자바 코드**를 압축하여 블랙보드에 지정된 시간까지 반드시 제출하십시오.)
Final Exam: 16:00 PM ~ 17:40 PM (100 Min.)
*I highly recommend that you start submitting to the Blackboard at 17:35PM, at least 5 minutes before the end. If you do not submit by the given time, you will receive 0 points.
(시험 끝나기 5분전인 5시 35분에는 블랙보드에 제출을 시작하길 권장합니다. 5시 40분에는 제출할 수 없도록 닫히며, 제출을 못했을 시 당연 0점 처리됩니다.)

(3) The Java code should be executable when the TA runs. The submitted compressed file (.zip) must be named Final_YourID_Yourname.zip.
(자바 코드는 TA가 돌렸을 시 깔끔하게 돌아가야 하며, 제출될 압축파일은 반드시 Final_YourID_Yourname.zip 로 명명하여 제출하십시오.)

(4) There is a partial score. Even if you can't resolve it completely, I hope you can try it as far as you can.
(부분 점수가 있으므로, 완벽하게 해결하지 못하더라도 할 수 있는 만큼 시도해 보길 바랍니다.)

(5) There are 4 questions in total, but I don't expect you to resolve all the questions within the given time. I hope you select the problem you can solve and solve it quickly. "Choice and focus" are also one of the programmer's core abilities. If you've followed the class normally and know your proficiency, I'm sure you'll do it well.
(총 4개의 질문이 있지만, 이것을 제한 시간내에 모두 해결할 수 없을 것입니다. 본인의 수준을 고려했을 때, 풀 수 있는 문제를 제한시간내에 가급적 많이 해결하십시오. "***선택과 집중***"은 프로그래머의 핵심 능력 중 하나입니다. 수업을 정상적으로 따라왔으며, 자신의 프로그래밍 숙달도를 알고 있는 학생이라면, 이를 잘 해낼 수 있을 것입니다.)

(6) To prevent abuse of ChatGPT, pattern recognition for issues is left to the developer role in most questions. Through this opportunity, I want you to directly feel in what areas you can have an edge over other developers in the AI era. "Raise an issue/Pattern Recognition/Requirement Segment" etc. ChatGPT is still far from perfect and can't beat a human developer. In other words, at this moment when everyone can use ChatGPT, the era has arrived where it is also very important who can efficiently divide and ask questions first. Fuse your abilities and technological advancements to show your best performance!
(ChatGPT의 남용을 방지하기 위해 대부분의 질문에서 문제에 대한 패턴 인식은 개발자 역할에 맡겨졌습니다. AI시대에서 다른 개발자에 비해 여러분이 어떤 부분에서 우위에 설 수 있는지를 이번 기회를 통해 직접적으로 느껴보기를 원합니다. "문제 제기 / 패턴 인식 / 요구사항 분할 능력" 등은 ChatGPT가 아직까지 인간 개발자를 뛰어 넘을 수 없습니다. 즉, 모두가 ChatGPT를 사용할 수 있는 이 순간에, 누가 먼저 효율적으로 문제를 분할하여 질문 할 수 있는지 여부도 매우 중요한 시대가 도래했습니다. 당신의 능력과 기술 발전을 융합하여 최고의 성과를 보여주십시오!)

(Q1) Please implement a very simple chatbot by introducing a Server-Client model. **The given "Chatbot.txt" (pictured below) is a file matched for appropriate responses** when a user asks a question. Of course, actual ChatGPT has a very large amount of response data compared to this, but the basic operating mechanism of chatbots is quite similar. **The data in that file is separated by tabs**. Please **design the UI of the client and server freely**, but be sure to evaluate whether the server side can give a natural answer when a user asks a question.**(25 Points).**

(Server-Client model을 도입하여 매우 간단한 Chatbot을 구현하십시오. 주어진 Chatbot.txt 은 (아래그림) 사용자가 질문을 했을 시 적절한 응답에 대해 매치되어 있는 파일입니다. 물론 실제 ChatGPT는 이것에 비해 매우 방대한 답변 방식 데이터를 지니고 있지만, 챗봇의 기본적인 작동 매커니즘은 상당히 유사합니다. 해당 파일의 자료는 tap으로 구분되어 있습니다. 클라이언트와 서버의 UI는 자유롭게 디자인하되, 사용자가 질문할 때 서버측에서 자연스러운 답변을 줄 수 있는지를 반드시 평가해야 합니다.)

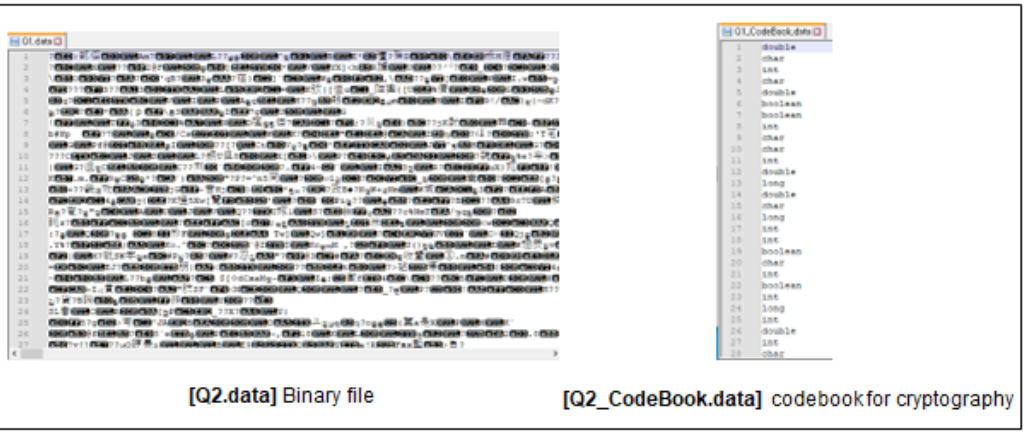
Question	Answer
hi, how are you doing?	i'm fine. how about yourself?
i'm fine. how about yourself?	i'm pretty good. thanks for asking.
i'm pretty good. thanks for asking.	no problem. so how have you been?
no problem. so how have you been?	i've been great. what about you?
i've been great. what about you?	i've been good. i'm in school right now.
i've been good. i'm in school right now.	what school do you go to?
what school do you go to?	i go to pcc.
i go to pcc.	do you like it there?
do you like it there?	it's okay. it's a really big campus.
it's okay. it's a really big campus.	good luck with school.
good luck with school.	thank you very much.
how's it going?	i'm doing well. how about you?
i'm doing well. how about you?	never better, thanks.
never better, thanks.	so how have you been lately?
so how have you been lately?	i've actually been pretty good. you?
i've actually been pretty good. you?	i'm actually in school right now.
i'm actually in school right now.	which school do you attend?
which school do you attend?	i'm attending pcc right now.
i'm attending pcc right now.	are you enjoying it there?
are you enjoying it there?	it's not bad. there are a lot of people there.
it's not bad. there are a lot of people there.	good luck with that.
good luck with that.	thanks.
how are you doing today?	i'm doing great. what about you?
i'm doing great. what about you?	i'm absolutely lovely, thank you.
i'm absolutely lovely, thank you.	everything's been good with you?
everything's been good with you?	i haven't been better. how about yourself?
i haven't been better. how about yourself?	i started school recently.
i started school recently.	where are you going to school?
where are you going to school?	i'm going to pcc.

[Chatbot.txt]

(Q2) Please develop a decryption program in the form of a server-client model that can decrypt the binary file given below based on the format described in the codebook. Binary files can be restored only by the order of the given codebook, and the following **two results should be obtained**. **The first is to get the same result as the screenshot** to check if the restore was successful. **Second, separate each different form, save it in a human readable file format, respectively**, and submit these files (5 files). On the client side, only the UI to pass only the query binary files to the server side and the function to show the result are required. All other codebook information and the decoding process must be processed on the server side, and the result must be delivered to the client. The UI design for the screen that submits the file and shows the result is free form. **(25 Points).**

(아래와 같이 주어진 Binary file을 codebook의 형식을 바탕으로 해독할 수 있는 복호 프로그램을 서버-클라이언트 형식으로 개발하십시오. 바이너리 파일은 주어진 코드북의 순서에 의해서만 복원될 수 있으며, 아래와 같은 두 가지 결과물을 얻어야 합니다. 첫번째는 스크린샷과 동일한 결과를 얻어서 복원이 성공했는지를 확인하십시오. 두번째는 각각의 서로 다른 양식을 각각 분리하여 사람이 읽을 수 있는 파일 형식으로 저장하고 파일들을 제출하십시오 (총 5개의 파일). 클라이언트 측에서는 쿼리 바이너리 파일만 서버 측으로 넘기는 UI와 그 결과를 보여주는 기능이 필요합니다. 다른 모든 코드북 정보와 디코딩 프로세스는 서버 측에서 처리되어야 하며 결과는 클라이언트로 전달되어야 합니다. 파일을 제출하고, 결과를 보여주는 화면에 대한 UI design은 자유 형식입니다.)

<Input>



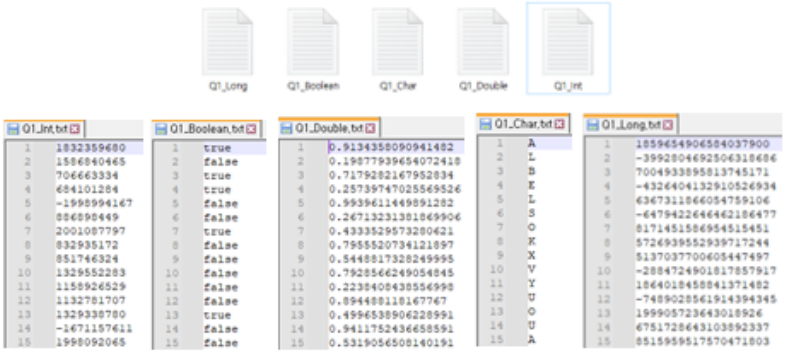
[Q2.data] Binary file

[Q2_CodeBook.data] codebook for cryptography

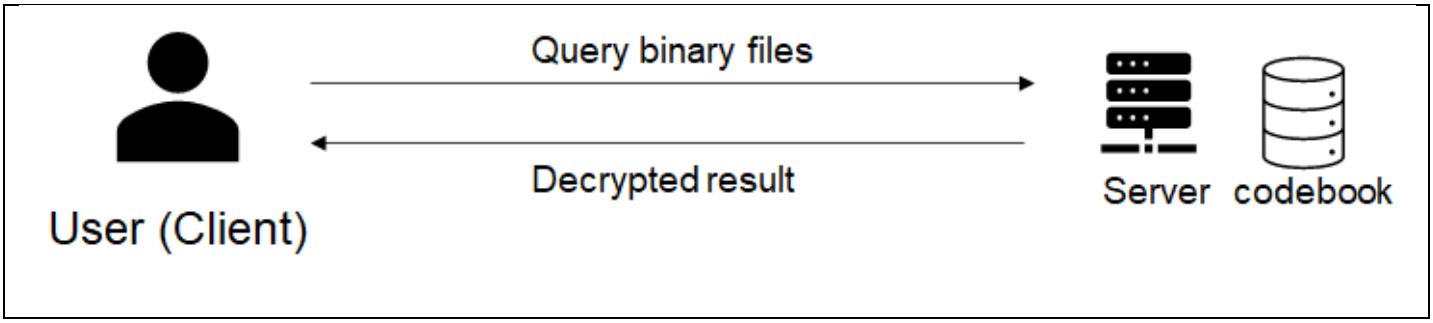
<Output>

```
The sum of all data of type double as a result of decryption is: 5986.477978998308
============================================================
# of integer values: 11939
# of boolean values: 11877
# of double values: 11955
# of char values: 11875
# of long values: 11836
```

[Outcome 1] To check if the decryption was successful, print the sum of all double values and the number of values of each type as above.



[Outcome 2] Write each type of value separately in human-readable text format to a separate files as above.



[Result Screenshot]

(Q3) In the decryption program developed in (Q2), if the user inputs 1000 query files, we want to implement a parallel computing algorithm on the server side that processes them simultaneously, rather than sequentially decrypting 1000 files. **Please improve the algorithm on the server side to make this possible**, and **compare the runtime of the parallelized version and the existing version for the given 5 binary file inputs**. You can experiment with 5 identical files. **(25 Points).**

((Q2) 에서 개발된 복호화 프로그램에서, 만약 사용자가 1000개의 쿼리파일을 입력했다면, 순차적으로 1000개의 파일을 복화 하는 것이 아니라, 이를 동시에 처리하게 만드는 병렬 컴퓨팅 알고리즘을 서버 측에 구현하고자 합니다. 이것이 가능하도록 Server측의 알고리즘을 개선하고, 병렬화 버전과 기존버전의 runtime 비교를 주어진 5개의 binary 파일 입력에 대해서 실시하십시오. 동일파일 5개로 실험해도 됩니다.)

[Result Screenshot & Runtime Comparison]

(Q4) We want to create a used car management program. However, unlike midterm exam, all computation must be performed using a server-client model that can be handled on the server side. **Write Automobile & Q4.java main classes** that satisfies the following requirements **(25 Points):**

(Kor. 중고차 관리 프로그램을 만들고자 합니다. 다만, 중간고사와 달리 모든 계산은 서버 측에서 처리할 수 있는 서버-클라이언트 모델을 사용하여 수행해야 합니다. 아래의 요구조건을 만족할 수 있도록 Automobile 클래스와 Q4 메인 클래스를 구현 하십시오).

	A	B	C	D	E	F	G	H	I	J	K	L
1	Model	Miles/(US	Number c	Displacem	Gross hor	Rear axle	Weight (1	1/4 mile t	Engine (0	Transmiss	Number c	Number c
2	Mazda RX4	21	6	160	110	3.9	2.62	16.46	0	1	4	4
3	Mazda RX4 Wag	21	6	160	110	3.9	2.875	17.02	0	1	4	4
4	Datsun 710	22.8	4	108	93	3.85	2.32	18.61	1	1	4	1
5	Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	1	0	3	1
6	Hornet Sportabout	18.7	8	360	175	3.15	3.44	17.02	0	0	3	2
7	Valiant	18.1	6	225	105	2.76	3.46	20.22	1	0	3	1
8	Duster 360	14.3	8	360	245	3.21	3.57	15.84	0	0	3	4
9	Merc 240D	24.4	4	146.7	62	3.69	3.19	20	1	0	4	2
10	Merc 230	22.8	4	140.8	95	3.92	3.15	22.9	1	0	4	2
11	Merc 280	19.2	6	167.6	123	3.92	3.44	18.3	1	0	4	4
12	Merc 280C	17.8	6	167.6	123	3.92	3.44	18.9	1	0	4	4
13	Merc 450SE	16.4	8	275.8	180	3.07	4.07	17.4	0	0	3	3
14	Merc 450SL	17.3	8	275.8	180	3.07	3.73	17.6	0	0	3	3
15	Merc 450SLC	15.2	8	275.8	180	3.07	3.78	18	0	0	3	3
16	Cadillac Fleetwood	10.4	8	472	205	2.93	5.25	17.98	0	0	3	4
17	Lincoln Continental	10.4	8	460	215	3	5.424	17.82	0	0	3	4
18	Chrysler Imperial	14.7	8	440	230	3.23	5.345	17.42	0	0	3	4

(Step 1) Implement an Automobile class that can load information from a Q4.Data (tab-delimited) file and turn it into an individual object. All parts of a given material must be reflected in the object. (Kor. Q4.Data(탭으로 구분) 파일에서 정보를 불러와서 이를 개별 객체로 전환할 수 있는 Automobile 클래스를 구현하십시오. 주어진 자료의 모든 부분이 객체에 반영되어야 합니다.)

(Step 2) Implement a constructor that allows developers to create additional car objects from code. At this time, actively utilize at least 3 overloading constructors so that if a specific value is not known, it can be treated as a "NA" value. (Kor. 개발자가 코드에서부터 추가적으로 자동차 객체를 생성할 수 있는 생성자를 구현하십시오. 이 때, 특정한 값이 알려지지 않았다면 "NA" 값으로 처리될 수 있도록 최소 3개의 오버로딩 생성자를 적극적으로 활용하십시오.)

(Step 3) Randomly generate 100 new cars from the main class Q4. It's okay to give a completely random car model name (duplication is not allowed). Implement the genAutomobile() method so that all other feature values can be randomly generated within the existing data range in Q4.data file.
(Kor. 메인 클래스 Q4에서 100대의 새로운 자동차를 랜덤하게 생성하십시오. 자동차 모델명은 완전하게 랜덤하게 지어도 무방합니다 (중복은 불허합니다). Q4.data 파일의 기존 데이터 범위 내에서 다른 모든 특성 값이 생성될 수 있도록 genAutomobile() 함수를 구현하십시오.)

(Step 4) Please implement that program as Server-Client model. All used car status data must exist on the server side, and only the UI for registering/removing/viewing vehicles must exist on the user side.
(Kor. 해당 프로그램을 Server-Client model으로 구현하십시오. 모든 중고차 현황 자료는 서버측에 존재하여야 하며, 유저쪽에는 차량을 등록/제거/열람할 수 있는 UI만 존재해야 합니다.)

[Result Screenshot]

-At the end- Well done!