**파이썬 중간고사**

기간: 10/27 09:00 ~ 11/03 09:00 (기간엄수)

업로드 파일:

* 실행화면이 담긴 레포트 (**각 문제별로 1~2장 내외 분량**으로 작성), 레포트에는 **실행화면**, **핵심코드**, **구현방식에 대한 간단한 설명**이 반드시 들어가야함.
* 전체 소스코드 (**문제별로 폴더를 생성하여 하나의 압축파일**로 제출)
* **부정행위 적발 시 0점 처리 (유사한 코드 발견 시)**

**문제 1**. 1에서 x까지의 정수 중에 4또는 8의 배수가 아닌 수를 모두 리스트에 담고 출력하시오. **(10점)**

* x는 숫자를 입력받는다.
* 반복문을 사용하여 구현한다 (5점)
* x를 매개변수로한 함수의 형태로 구현한다 (5점)
* 결과로 나온 리스트의 모든 숫자의 총 합을 출력한다 **(보너스 점수 1점)**

**문제 2**. 히브리어로 비밀 문서를 작성했던 것에서 유래한 아트배쉬(Atbash) 암호법이 있음. 이 방법은 아래 그림과 같이 알파벳의 숫서를 뒤바꾸는 방법을 사용한다. 여기에 대문자와 아라비아 숫자도 같은 방법으로 암호화한다고 하고, 입력받은 문자열을 암호화하고, 그 결과를 다시 복호화하는 프로그램을 작성해라. **(20점)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 소문자 | a | b | c | d | … | z |
| 암호화 | z | y | x | w | … | a |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 대문자 | A | B | C | D | … | Z |
| 암호화 | Z | Y | X | W | … | A |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 숫자 | 0 | 1 | 2 | 3 | … | 9 |
| 암호화 | 9 | 8 | 7 | 6 | … | 0 |

**(주의) 알파벳과 숫자가 아닌 문장이 입력되었을 경우 오류메시지를 출력한다.**

|  |
| --- |
| 출력 예시1)  평문을 입력하세요 >> adz119  암호화된 문장은 >> zwa880  입니다! |
| 출력 예시2)  평문을 입력하세요 >> 안녕하세요?  입력은 알파벳과 숫자로만 입력하세요!! |

**문제 3**. U(Up), D(Down), L(Left), R(Right)로 움직일 수 있는 로봇이 좌표 (0,0)에 있다. 이 로봇에 대한 이동명령이 방향과 거리의 문자열로 주어질 때 이 명령을 해독하여 현재 좌표를 알려주는 함수 Current\_position(command) 함수를 정의하여라. 각 명령의 구분은 콤마로 구별한다. **(20점)**

**(주의) 알수 없는 명령이 존재할 시 오류메시지를 출력한다.**

|  |
| --- |
| 출력 예시1)  명령을 입력하세요 >> U100, D170, L120, U50, D10, R20  로봇의 좌표는 현재: -100, -30 입니다. |
| 출력 예시2)  명령을 입력하세요 >> K100, D170, L120, U50, D10, R20  알수 없는 명령이 존재합니다. |

**문제 4**. 리스트 자료형을 활용하여 자료구조 큐 (Queue)를 구현하며 아래와 같은 세부 요구사항을 구현하시오. 큐는 먼저 넣은 데이터가 먼저 나오는 First In First Out (FIFO)구조로 저장하는 형식을 말한다. (참고: [큐 (자료 구조) - 위키백과, 우리 모두의 백과사전 (wikipedia.org)](https://ko.wikipedia.org/wiki/%ED%81%90_(%EC%9E%90%EB%A3%8C_%EA%B5%AC%EC%A1%B0))) **(20점)**

* 큐의 맥시멈 크기를 입력 받는다.
* 큐에서의 명령 2가지 (입력 x, 출력)을 입력받으며 exit를 입력하면 프로그램 종료된다.
* 큐에 데이터를 집어 넣는 data\_In(x) 함수를 구현한다. **(10점)**
* 큐에서 데이터를 꺼내오는 함수 data\_pop() 함수를 구현한다. . **(10점)**
* 만약 큐에 데이터가 꽉 찻을 때 데이터를 집어넣을 경우 오버플로우 메시지를 출력한다.
* 만약 큐에 데이터가 없을 때 데이터를 꺼내려고 하면 언더플로우 메시지를 출력한다

|  |
| --- |
| 출력 예시1)  생성할 큐의 크기를 설정해 주세요 >> 5  명령을 입력하세요 >> 입력 10  현재 큐 현황: [10]    령을 입력하세요 >> 입력 4  현재 큐 현황: [10,4]    령을 입력하세요 >> 입력 22  현재 큐 현황: [10, 4, 22]    령을 입력하세요 >> 입력 9  현재 큐 현황: [10, 4, 22, 9]    명령을 입력하세요 >> 입력 1  현재 큐 현황: [10, 4, 22, 9, 1]  명령을 입력하세요 >> 입력 55  오버플로우 입니다! 큐가 꽉찼습니다.    명령을 입력하세요 >> 출력  현재 큐 현황: [4, 22, 9, 1] 출력된 데이터: 10  명령을 입력하세요 >> 출력  현재 큐 현황: [22, 9, 1] 출력된 데이터: 4  명령을 입력하세요 >> 출력  현재 큐 현황: [9, 1] 출력된 데이터: 22  명령을 입력하세요 >> 출력  현재 큐 현황: [1] 출력된 데이터: 9  명령을 입력하세요 >> 출력  현재 큐 현황: [] 출력된 데이터: 1  령을 입력하세요 >> 출력  언더플로우 입니다! 큐가 꽉찼습니다. |

문제 5. Tic Tac Toe 게임을 다음과 같이 설계한다 **(30점)**

(참고: [틱택토 - 나무위키 (namu.wiki)](https://namu.wiki/w/%ED%8B%B1%ED%83%9D%ED%86%A0))

* 설계하는 게임은 플레이어와 컴퓨터(AI)가 대전하는 형식으로 설계한다.
* 프로그램을 실행하면 3x3의 판을 그려준다.
* 플레이어는 무조건 선(先)으로 플레이하며 놓을 x와 y의 좌표를 입력한다.
* 그 이후, 컴퓨터는 남은 공간 중 랜덤한 좌표를 골라서 놓는다.
* 자신만의 컴퓨터 AI 알고리즘을 구현한 경우 **보너스점수 10점** (레포트에 알고리즘 원리 반드시 설명 필수!)
* 순서를 반복하다가 한방향 빙고를 완성하는 순간 승패 여부를 출력한다.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 출력 예시)   |  |  |  | | --- | --- | --- | |  |  |  | |  |  |  | |  |  |  |   >> 놓을 곳의 좌표를 입력하세요: 2, 2   |  |  |  | | --- | --- | --- | |  |  |  | |  | O |  | |  |  |  |   >> 컴퓨터가 놓을 차례입니다.   |  |  |  | | --- | --- | --- | |  | X |  | |  | O |  | |  |  |  |   >> 놓을 곳의 좌표를 입력하세요: 1, 1   |  |  |  | | --- | --- | --- | | O | X |  | |  | O |  | |  |  |  |   >> 컴퓨터가 놓을 차례입니다.   |  |  |  | | --- | --- | --- | | O | X |  | |  | O | X | |  |  |  |   >> 놓을 곳의 좌표를 입력하세요: 3, 3   |  |  |  | | --- | --- | --- | | O | X |  | |  | O | X | |  |  | O |   >> 축하합니다! 당신의 승리입니다! |