**REPORT**

(프로그래밍 과제 3)

로고, 상징, 등록 상표, 폰트이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

|  |  |
| --- | --- |
| 이름 | 김민서 |
| 학과 | 컴퓨터인공지능공학부 |
| 학번 | 202213083 |
| 과목명 | 이산수학 |
| 교수님 | 신상욱 교수님 |
| 분반 | 101분반 |
| 제출일 | 2025.05.07 |

**o 정렬 알고리즘과 이진 탐색 알고리즘 구현**

**- 원소의 개수인 양의 정수 n을 입력받는다.**

**- n개의 양의 정수를 입력받는다 (문제를 간단히 하기 위해 중복된 값을 입력하지 않는다).**

**- 삽입정렬 알고리즘을 통해 오름차순으로 정렬된 원소들의 리스트를 출력한다.**

**- 검색할 값을 입력받는다.**

**- 이진 탐색을 통해 찾고자 하는 값의 저장 위치를 출력한다. 없는 경우에는 "0" 또는 "없음"이라고 출력한다.**

**1. [3장 강의 자료 "Ch03\_Algorithms-p1.pdf" 15~16페이지] Insertion Sort (삽입 정렬) 구현**

**2. [3장 강의 자료 "Ch03\_Algorithms-p1.pdf" 9~10페이지] Binary Search(이진 탐색) 구현**

**===> (참고) 아래 실행 예와 유사하게 동작하는 프로그램 구현:**

**텍스트, 스크린샷, 폰트이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.**

**# 코드**

# 원소의 개수 입력 및 리스트 생성

N = int(input("원소의 개수 : "))

arr = [0 for i in range(N)]

# N개의 양의 정수 입력

for i in range(N):

    arr[i] = int(input())

# 삽입 정렬 알고리즘 구현

for j in range(1, N):

    i = 0

    # 알맞은 위치 찾기

    while arr[j] > arr[i]:

        i += 1

    m = arr[j]

    # 오른쪽으로 한 칸씩 이동

    for k in range(j, i, -1):

        arr[k] = arr[k - 1]

    # m을 올바른 위치에 삽입

    arr[i] = m

# 삽입정렬 이후 리스트 출력

print("정렬된 리스트 : ", arr)

# 검색할 값 입력

Find = int(input("검색할 원소 : "))

# 이진 탐색

p = 0            # 왼쪽 끝 인덱스

r = N - 1        # 오른쪽 끝 인덱스

while p < r:

    # 범위 내 중앙에 위치한 값

    q = (p + r) // 2

    # 크기 비교 후 범위 재지정

    if (Find > arr[q]):

        p = q + 1

    else:

        r = q

# 찾고자 하는 값의 저장 위치 출력

if p < N and arr[p] == Find:

    print("저장된 위치 :", p + 1)

else:

    print("0")

**# 실행 결과**

**텍스트, 스크린샷, 디스플레이, 소프트웨어이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.**

**2. [2장 강의 자료 "Ch02\_Set-Function-Sequence-Matrix-p2.pdf" 28페이지]  의사난수 생성 프로그램 구현**

**- 모듈러(modulus) m, multiflier a, increment c, seed s을 입력받는다.**

**- 생성할 난수의 개수 n을 입력받는다.**

**- n개의 난수를 출력한다.**

**# 코드**

# 초기값 입력

m, a, c, s = map(int, input("모듈러(modulus), multiflier, increment, seed를 입력 : ").split())

# 생성할 난수의 개수 입력

num = int(input("생성할 난수의 개수 : "))

list = [0 for i in range(num)]

for i in range(num):

    # 예외처리) 첫 번째 값(list[0])은 seed 값

    if (i == 0):

        list[0] = s

    # 예제의 난수 생성 공식에 따라 값 계산 후 저장

    else:

        list[i] = (a \* list[i - 1] + c) % 11

# 출력

print("생성한 난수 : ", list)

**# 실행 결과**

**텍스트, 스크린샷, 폰트이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.**