**REPORT**

(프로그래밍 과제 1)

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 | 2025.03.27 |

**1. 집합 X와 집합 Y를 나타내는 배열 A와 B가 주어졌을 때, 두 집합의 합집합, 교집합, 차집합을 구하는 프로그램을 작성하시오.**

* **n개의 원소를 가진 집합은 적어도 (n+1)의 크기를 갖는 배열로 표현된다고 가정한다.**
* **문제를 간단하게 하기 위해, 집합의 원소들은 1~100 사이의 정수들 중의 임의의 원소들로 구성될 수 있다고 가정한다. 또한 중복되는 원소는 없는 것으로 가정한다.**
* **집합의 원소들은 배열에서 첫번째 위치부터 시작하여 연속적으로 배치되며, 0으로 끝난다고 가정한다. 즉, 0을 포함하는 집합은 없는 것으로 가정한다. 0은 집합에서 원소가 더 이상 존재하지 않는다는 것을 나타낸다.**
* **집합의 크기와 집합의 원소들은 사용자로부터 입력 받도록 한다.**

**# 코드**

* # 리스트를 입력 받는 함수
* # 원소들을 입력 후, 0을 입력하면 집합 입력 종료
* def get\_set():
* arr = list(map(int, input("집합 입력 (0으로 종료) : ").split()))
* return arr[:-1]  # 0을 제외한 부분을 리스트로 반환
* # 합집합 함수
* # result 리스트 안에 A, B에 해당하는 원소가 없으면 추가가
* def union(A, B):
* result = []
* for x in A:
* if x not in result:
* result.append(x)
* for y in B:
* if y not in result:
* result.append(y)
* return sorted(result)
* # 교집합 함수
* # A, B 모두에 해당하는 원소만 result 리스트에 추가
* def intersection(A, B):
* result = []
* for x in A:
* if x in B and x not in result:
* result.append(x)
* return sorted(result)
* # 차집합 함수
* # A집합에는 포함되나, B집합에는 포함되지 않는 원소만 result 리스트에 추가
* def difference(A, B):
* result = []
* for x in A:
* if x not in B:
* result.append(x)
* return sorted(result)
* # 배열 입력 받기
* A = get\_set()
* B = get\_set()
* # 출력
* print("합집합 : ", union(A, B))
* print("교집합 : ", intersection(A, B))
* print("차집합 (A - B) : ", difference(A, B))
* print("차집합 (B - A) : ", difference(B, A))

**# 실행 결과**

**텍스트, 스크린샷, 폰트, 디자인이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.**

**2. 집합 X를 나타내는 배열 A가 주어졌을 때, 집합 X의 모든 부분집합을 구하는 프로그램을 작성하시오. 즉 집합 X의 멱집합(power set)을 생성하는 프로그램을 작성하시오.**

* **n개의 원소를 가진 집합 X가 적어도 (n+1)의 크기를 갖는 배열 A로 표현된다고 가정한다.**
* **집합의 원소들은 0 보다 큰 양의 정수들 중 임의의 원소들로 구성될 수 있다고 가정한다.즉, 1, 2, 3, 4, .... 들 중에서 임의의 원소들로 구성될 수 있다. 또한 중복되는 원소는 없는 것으로 가정한다.**
* **집합의 원소들은 배열에서 첫번째 위치부터 시작하여 연속적으로 배치되며, 0으로 끝난다고 가정한다. 즉, 0을 포함하는 집합은 없는 것으로 가정한다. 0은 집합에서 원소가 더 이상 존재하지 않는다는 것을 나타낸다.**
* **집합의 크기와 집합의 원소들은 사용자로부터 입력 받도록 한다.**

**# 코드**

* # 리스트 입력 함수
* # 원소들을 입력 후, 0을 입력하면 집합 입력 종료
* def get\_input\_set():
* A = list(map(int, input("집합의 원소를 공백으로 구분하여 입력 (마지막에 0 입력): ").split()))
* return A[:-1]  # 마지막 0 제거
* # 멱집합 생성
* def generate\_power\_set(arr):
* n = len(arr)
* power\_set = []
* for i in range(1 << n):  # 원소의 개수는 2^n 개(부분집합)
* subset = []
* for j in range(n):
* if i & (1 << j):  # j번째 원소가 부분집합에 포함되는지 확인
* subset.append(arr[j])
* power\_set.append(subset)
* return power\_set
* # main 함수
* def main():
* input\_set = get\_input\_set() # 전체 원소 입력
* power\_set = generate\_power\_set(input\_set) # 멱집합 생성
* print("\n멱집합:")
* for subset in power\_set:
* print(subset)
* if \_\_name\_\_ == "\_\_main\_\_":
* main()

**# 실행 결과**

**텍스트, 스크린샷, 디스플레이, 소프트웨어이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.**