〈C프로그래밍 및 실습〉 12주차 실습문제

※ 문제에 대한 안내

- 출력 예시에서 □는 공백을 의미한다.
- 입출력 예시에서 → 이 후는 각 입력과 출력에 대한 설명이다.

[문제 1] (100점) 정수 N을 입력받는다. (함수 사용금지)

- (1) N개의 정수를 입력받아 <u>0보다 큰 정수만 배열에 저장</u>하고, 배열에 저장된 정수의 개수를 출력한다. 단, N≦100이고, 입력되는 정수 중에 한 개 이상의 양의 정수가 있다고 가정한다.
- (2) <u>배열에 저장된 정수의 개수가 홀수</u>이면, 배열에 저장된 정수를 작은 수부터 큰 수까지 <u>오름차순으로</u> 정렬하여 출력한다.
- (3) <u>배열에 저장된 정수의 개수가 짝수</u>이면, 배열에 저장된 정수를 큰 수부터 작은 수까지 <u>내림차순으로</u> <u>정렬</u>하여 출력한다.
- (4) 정렬된 배열에서 <u>홀수가 연속해서 나오는 횟수를 세어서, 이 중 최대 횟수를 출력</u>한다. 예를 들어, 7 9 11과 같이 홀수 3개가 배열에 연속해서 저장된 경우, 홀수가 연속해서 나오는 횟수는 3회로 센다. 배열에 저장된 정수 중 한 개 이상의 홀수가 있다고 가정한다.

입력 예시 1

16 → N 5 1 -2 3 19 6 7 8 17 -4 9 11 -6 16 13 15

출력 예시 1

13	→ 배열에 저장된 정수의 개수
	→ 홀수이므로 오름차순 정렬→ 최대연속 홀수는 4개
4	'/ 띄네한국 출구는 왕

[문제 2] (100점) 정수 N과 M을 입력받고, N개의 양의 정수를 입력받아 배열에 저장한다. 단, 1≦N ≤100이고, 1≦M≤N이다.

- (1) 배열에 저장된 <u>정수를 앞에서부터 M개씩 묶어</u>서, <u>각 묶음의 평균값보다 크거나 같은 정수를 차례로</u> 출력한다.
- (2) 배열에 저장된 정수를 앞에서부터 M개씩 묶어서, 각 묶음의 두 번째로 큰 수를 차례로 출력한다.
- (3) 배열의 원소를 M개씩 묶고, 마지막 남은 원소의 개수가 M보다 작은 경우, 남은 원소의 평균보다 크 거나 같은 정수와 두 번째로 큰 수를 출력한다. 마지막 남은 원소가 한 개인 경우, 이 원소 값이 두 번째로 큰 값이 된다.

☞ 입력형식

- 첫 줄에 N과 M을 입력받는다.
- 두 번째 줄에 N개의 양의 정수를 입력받는다.

☞ 출력형식

- 첫 번째 줄에 각 묶음의 평균값보다 크거나 같은 정수를 출력한다.
- 두 번째 줄에 각 묶음의 두 번째로 큰 수를 출력한다.

입력 예시 1 출력 예시 1

6 4 → N, M	3 4 5 5	→ 평균보다 큰수
123455	3 5	→ 두 번째로 큰 수

입력 예시	2	출력	예시	2

10 3	20 30 50 60 80 90 100
<u>10 20 30</u> <u>40 50 60</u> <u>70 80 90</u> 100	20 50 80 100

[문제 3] (100점) 정수 N을 입력받고, N명의 학생들의 총점을 입력받아 등수와 학점을 계산하여 내림 차순으로 정렬하고 각 학생의 총점에 대해 등수와 학점을 출력하는 프로그램을 작성하라. 점수가 같은 경우, 입력된 순서를 유지하도록 한다.

- (1) 정수 N을 입력받고, N명의 학생의 총점을 입력받아 배열에 저장한다. 단, N≤20이다.
- (2) 학생의 총점을 배열에 저장한다.
- (3) 총점 순서대로 학생의 총점을 내림차순으로 정렬한다. 점수가 같은 경우, 입력된 순서를 유지한다.
- (4) 각 학생의 등수를 총점을 기준으로 계산한다. 단, 동점인 학생의 등수는 같으며, 같은 등수의 학생이 여러 명이면, 없는 등수가 있을 수 있다.
- (5) 총점을 기준으로 학점을 부여한다.
 - N명의 학생 중 상위 3<u>0% 이내의 학생에게 A 학점</u>을 부여한다.
 - 상위 70% 이내의 학생 중 A학점을 받지 못한 학생들에게 B학점을 부여한다.
 - A학점과 B학점을 부여받지 못한 나머지 학생들에게 F학점을 부여한다.
 - 같은 점수는 같은 학점부여 함.
- (6) 학생의 <u>총점, 등수, 학점을 출력한다. 출력 순서는 총점에 따라 내림차순으로 정렬</u>된 순서이다. (등수가 같은 경우 입력한 순서가 앞선 것을 먼저 출력함)

입력 예시 1 출력 예시 1

7 80 75 92 85 88 78 95	→ N → 총점	95 1 A 92 2 A 88 3 B 85 4 B 80 5 F 78 6 F	→ 상위30%는 2명: A→ 상위70%는 4명: B→ 나머지 F
		75 7 F	

[문제 4] (100점) 종료 조건까지 영어 대문자를 반복해서 입력받아, 배열 X에 저장한다.

- (1) <u>배열 X에 저장된 문자 중 중복된 문자를 제외하고 배열 Y에 저장</u>한 후, 배열 Y에 저장된 문자를 출력한다. 배열 X에 중복되어 나타나는 문자의 경우, 처음 나타난 문자를 배열 Y에 저장한다.
- (3) 배열 Y에 저장된 문자 중 <u>대문자-모음 또는 대문자-자음이 교대로 나오는 부분 배열을 찾는다</u>. 이러 한 <u>부분 배열 중에서 가장 긴 부분 배열의 원소를 출력</u>한다. 부분 배열 중에서 가장 긴 부분 배열의 크기를 가지는 부분 배열이 여러 개인 경우, 제일 처음 나온 부분 배열을 출력한다.
- (4) 만약 조건을 만족하는 부분 배열이 없으면 "none"을 출력한다.
 - 종료 조건 : 문자 '!' 입력 (문자 '!'는 배열에 저장하지 않는다.)
 - 입력되는 문자의 최대 개수는 100이다.

입력 예시 1 출력 예시 1

HELLOWORLD!	HELOWRD HELOW	→ 중복문자 제거하고 배열 Y에 저장
입력 예시 2	출력 예시 2	
XYZXYZXYZYZ!	XYZ none	→ 중복문자 제거하고 배열 Y에 저장

[문제 5] (100점) 주어진 단일 정수 N을 입력받아 역순으로 된 정수 M을 생성한다. 정수 M에 대해서 소수인지 확인하는 프로그램을 작성하라. 예를 들어, 입력 "12345"는 뒤집으면 "54321"이 되고, 이것이 소수인지 확인하는 것입니다.

<요구사항>:

- (1) 단일 정수의 최대 9자리 정수이다.
- (2) reverse_num 함수를 호출하여 정수를 뒤집는다.
- (3) is_prime 함수를 호출하여 뒤집은 정수가 소수인지 확인한다.
- (4) 원본 정수, 뒤집은 정수, 그리고 뒤집은 정수가 소수인지 여부를 나타내는 1 또는 0(소수이면 1, 소수가 아니면 0)을 main 함수에서 화면에 출력한다.
- (5) main 함수 위에 함수 원형 선언을 하고, 함수의 정의는 main 함수 아래에 오도록 작성한다.

<함수 설명>:

☞ reverse_num 함수

- 함수 원형: int reverse num(int num)
- 인자: 정수 num
- 반환 값: 입력된 정수를 뒤집은 값을 반환

☞ is_prime 함수

- 함수 원형: int is_prime(int num)
- 인자: 정수 num
- 반환 값: num이 소수이면 1, 소수가 아니면 0을 반환

입력 예시 1 출력 예시 1

12345	$\mapsto N$	54321 0	→ M을 출력 → M이 소수가 아니면 0

- ▶ 함수를 설명과 다르게 정의하거나, 목적에 맞지 않게 호출하여 사용한 경우 (함수 각각 50% 감점)
- ▶ 함수원형선언을 올바르게 하지 않거나, 함수를 main 함수 전에 작성한 경우 (각 함수 10%씩 감점)
- ▶ 전역변수를 사용한 경우 **(50% 감점)**