

## 2021 C 프로그래밍 과제 5

### 1. 현재 시간을 구하는 함수 제작(20)

`void split_time(long total_sec, int *hr, int *min, int *sec);`

`total_sec` 는 1970 년 1 월 1 일 이후 현재까지 흐른 시간을 전달하는 파라미터.

`hr, min, sec` 는 현재 시간을 저장하는 변수

`hr`:시간(0~23), `min`:분(0~59), `sec`:초(0~59) 데이터를 저장

출력은 반드시 `main()`함수에서 수행

조건, `time.h` 에서 `time(NULL)`만 사용가능, `localtime()`, `ctime()`, `strftime()` 등의 다른 함수는 사용 불가

결과는 UTC 시간으로 출력, 현지 시간으로 변경할 필요 없음

참고:

`time(NULL)`는 1970 년 1 월 1 일(UTC)부터 현재까지 흐른 시간을 초단위로 반환

UTC 시간 확인: <https://time.is/ko/UTC>

```
> ./main
Current UTC: 10시 13분 00초 입니다 .
> 
```

## 2. 1 차원 배열 오름차순 정렬 프로그램 만들기(40)

### a. 필요함수

```
void input_array(int [], int); // 배열에 데이터 입력 함수
void output_array(int [], int); // 배열 출력 함수
bool sort(int [], int); // 배열 정렬
void swap(int*, int*); //두 개의 데이터를 받아서 교환하는 함수
```

### b. int main()

배열의 크기를 입력 받고 배열 생성, 배열의 입력 함수 호출, 배열의 출력함수 호출  
sort() 함수 호출, 정렬된 배열 출력, 최대 최소 값 출력

### c. input\_array(): 생성된 배열에 10~100 사이의 임의의 값 입력

### d. output\_array(): 배열 출력

### e. sort(): 배열을 정렬하는 함수, swap 함수가 호출 되었으면 main()함수에서

“Called swap function” 메시지 출력을 판단할 수 있는 데이터를 return

### f. swap(): 두개의 정수 데이터의 메모리주소를 전달받아 두 데이터를 교환하는 함수

예시)

```
input number of data:30
Array:
 36 65 37 100 29 65 96 10 99 73 64 49 74 11 48 27 25 73 73 87 45 42 10 22 54 28 61 15 22 37
Called swap function
=====sorted array=====
Array:
 10 10 11 15 22 22 25 27 28 29 36 37 37 42 45 48 49 54 61 64 65 65 73 73 73 74 87 96 99 100
min: 10, max:100
```

```
input number of data:10
Array:
 31 21 59 40 39 50 41 21 78 68
Called swap function
=====sorted array=====
Array:
 21 21 31 39 40 41 50 59 68 78
```

### 3. 생성된 수의 빈도수 찾기(40)

- a. 사용자에게 데이터의 수(size: DATA array 의 크기)를 입력 받는다.
- b. 입력 받은 size 의 크기의 DATA array 생성: size 크기만큼의 정수형 배열(int array) 생성, rand()함수를 이용하여 0~10 사이의 수를 생성하여 각 element 에 입력한다. srand()함수를 이용하여 매번 프로그램실행(run)할 때마다 다른 sequence 의 값이 입력되도록 한다.
- c. 11 개의 int 를 저장할 수 있는 1-D array 를 생성하여 b 에서 생성한 수의 빈도수(frequency)를 입력한다.  
index 가 0 인 요소에 b 에서 생성된 수에서 0 의 개수 입력  
index 가 1 인 요소에 b 에서 생성된 수에서 1 의 개수 입력  
.  
.  
.  
index 가 10 인 요소에 b 에서 생성된 수에서 10 의 개수 입력
- d. 아래와 같이 Menu 를 화면에 출력한다.
  - (1) 생성된 데이터의 내용을 출력(display)
  - (2) 각 수의 빈도수(frequency)를 출력(1 차원 배열 출력)
  - (3) 입력된 data 의 표준편차 출력
  - (4) 입력 숫자를 DATA array 에서 찾기
  - (5) 최빈수(mode) 찾기
  - (6) 종료
- e. d. c 에서 출력한 메뉴를 선택하는 경우 다음과 같이 수행되도록 함수로 작성. 함수 호출 시 필요한 array 는 argument 로 호출한다.(모든 menu 의 수행을 '(6) 종료'를 제외하고 함수로 작성)
  - (1)의 경우 DATA array 의내용의 값(DATA array 의 element)을 한 줄에 10 개씩 (size/10+1)줄로 출력한다.
  - (2)의 경우 아래와 같은 format 으로 출력 한다. (단, 자리 수를 오른쪽으로 정렬)
  - (3)의 경우 표준 편차를 구한다(필요한 경우 math library 함수 사용)- 아래의 공식을 사용.  
표준편차는 분산의 제곱근(square root)를 구하면 됨
    - 분산(variance):  $v = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n-1}$   
xi: i 번째 값  
 $\bar{x}$ : 값의 평균
    - 표준편차(standard deviation):  $\sigma = \sqrt{v}$
  - (4) 0~10 사이의 숫자를 입력 받고 해당 숫자가 DATA array 에서 어디에 있는 지 index 출력
  - (5) 최빈수는 가장 많이 생성된 숫자를 의미하며 해당 숫자를 화면에 출력(display)
  - (6) 의 경우 전체 프로그램을 종료

- f. (6)종료를 선택하기 전까지는 프로그램이 계속 실행(사용자에게 메뉴를 출력하여 입력 받도록)되어야 한다.
- g. 함수 원형
- 데이터 생성 함수: `void input_data(int *, int *, int);`
  - 표준편차 함수: `float dev(int *,int);`
  - 최빈수 함수: `int fun5(int*);`
  - 그 외의 함수는 원하는 형태로 생성하면 됨
- h. 프로그램 실행 예시는 다음 페이지에서 확인할 수 있음

예시는 다음 페이지에 있음.

```

input number of data:20

*****
** (1) 생성된 데이터의 내용 출력(display) **
** (2) 각 숫자의 빈도수(frequency)를 출력 **
** (3) 입력된 data의 표준편차 출력 **
** (4) 입력 숫자를 DATA array에서 찾기 **
** (5) 최빈수(mode) 찾기 **
** (6) 종료 **
*****
input menu:1

8 8 10 10 4 2 10 9 1 10
2 3 7 0 2 1 9 10 6 6

*****
** (1) 생성된 데이터의 내용 출력(display) **
** (2) 각 숫자의 빈도수(frequency)를 출력 **
** (3) 입력된 data의 표준편차 출력 **
** (4) 입력 숫자를 DATA array에서 찾기 **
** (5) 최빈수(mode) 찾기 **
** (6) 종료 **
*****
input menu:2

00: 1 01: 2 02: 3 03: 1 04: 1 05: 0 06: 2 07: 1 08: 2 09: 2 10: 5

*****
** (1) 생성된 데이터의 내용 출력(display) **
** (2) 각 숫자의 빈도수(frequency)를 출력 **
** (3) 입력된 data의 표준편차 출력 **
** (4) 입력 숫자를 DATA array에서 찾기 **
** (5) 최빈수(mode) 찾기 **
** (6) 종료 **
*****
input menu:3

Standard deviation: 3.65

*****
** (1) 생성된 데이터의 내용 출력(display) **
** (2) 각 숫자의 빈도수(frequency)를 출력 **
** (3) 입력된 data의 표준편차 출력 **
** (4) 입력 숫자를 DATA array에서 찾기 **
** (5) 최빈수(mode) 찾기 **
** (6) 종료 **
*****
input menu:4

input find number:10
data[2] data[3] data[6] data[9] data[17]

*****
** (1) 생성된 데이터의 내용 출력(display) **
** (2) 각 숫자의 빈도수(frequency)를 출력 **
** (3) 입력된 data의 표준편차 출력 **
** (4) 입력 숫자를 DATA array에서 찾기 **
** (5) 최빈수(mode) 찾기 **
** (6) 종료 **
*****
input menu:5

Mode:10, 5번 생성됨

*****
** (1) 생성된 데이터의 내용 출력(display) **
** (2) 각 숫자의 빈도수(frequency)를 출력 **
** (3) 입력된 data의 표준편차 출력 **
** (4) 입력 숫자를 DATA array에서 찾기 **
** (5) 최빈수(mode) 찾기 **
** (6) 종료 **
*****
input menu:6

Good Bye

```