Problem 1.

Complete get_kth_largest() in the given source code.

int get_kth_largest(int *nums, int k) : Finds the k-th largest element in an integer array pointed to by *nums*. All values of the array are non-negative except the last one (END_MARK : -1) to indicate the end of the array.

- Parameters
  - nums - pointer to the END_MARK (-1) terminated integer array to be searched for.
  - k - the order that we want to find.
- Return value
  - Returns the value of the k-th largest element
  - If k is smaller than 1 or larger than the number of elements in the array, return -1;

You can add new functions and variables.

```c
#include <stdio.h>
#include <stdlib.h>

#define END_MARK  -1
#define MAX_SIZE  10000

int get_kth_largest(int *nums, int k) {

  return -1;
}
```

Problem 2.

Complete the **print_local_time_after_kdays(time_t t, int k)** printing out the local time (from the machine's perspective) after "k" days of the given time "t" using the localtime() library function.

Note that the result of the localtime() may differ from what you expect. The localtime of the Goorm's VM may be different from ours.

(주어진 시간 t에서 k 일 이후의 시간을 출력하는 함수 print_local_time_after_kdays(time_t, int k)를 표준 라이브러리 함수 localtime()을 이용하여 완성하라. localtime은 프로그램이 동작하는 기계 기준이다. 단 구름의 가상기계(VM)의 localtime이 우리와 다를 수 있음에 유의하라. 출력 형식은 아래와 같다)

Output format example:

2022/12/1, 01:06:32, Thursday

```c
#include <stdio.h>
#include <time.h>

void print_local_time_after_kdays(time_t t, int k) {


  return;
}

int main(void) {
  int k=0;
  time_t now = 1669856792; // 2022/12/01 01:06:32

  scanf("%d", &k);

  print_local_time_after_kdays(now, k);

  return 0;
}
```

Problem 3.

Complete the **stud_bubble_sort(STUD *pnucse)** sorting an array of STUD type whose address is given as pnucse in the decreasing order of points using **the bubble sort algorithm**.

You are not allowed to add a new function.

You can use the following functions whose source codes are hidden on purpose.

- void stud_print(STUD *ps);
- void stud_swap(STUD *a, STUD *b);
- STUD * stud_get_last(STUD *ps_array);
- int stud_compare_points(STUD *ps1, STUD *ps2);

```c
#include <stdio.h>
typedef struct student {
  int id;
  char *pname;
  double points;
} STUD;

void stud_print(STUD *ps);
void stud_swap(STUD *a, STUD *b);
STUD * stud_get_last(STUD *ps_array);
int stud_compare_points(STUD *ps1, STUD *ps2);

void stud_bubble_sort(STUD * pnucse) {

  return;
}


int main(void) {
  STUD pnucse[] = { {1, "Choi", 9.9}, {2, "Park", 0.1},
    {3, "Kim", 5.0 }, {4, "Lee", 3.0 }, {5, "Moon", 9.5 },
    {6, "Kang", 7.0 }, {7, "Jeon", 0.9 }, {-1, NULL, 0 } };

  STUD * ps_cur = pnucse;
  int test_id = 0;
  scanf("%d",&test_id);
  if (test_id) set_values_of_pnucse(pnucse, test_id);

  stud_bubble_sort(pnucse);

  while (ps_cur->id > 0)
    stud_print(ps_cur++);

  return 0;
}
```