3. RAND_MAX (32767) 보다 더 큰 값의 Big Rand 난수 배열 생성, 선택 정렬 및 출력 (30점, 45분)

1) BigArray_Algorithms.cpp와 BigArray_Algorithms.h 파일을 준비
2) BigArray_Algorithms.cpp 파일에는 큰 규모의 배열 데이터를 처리하기 위한 함수들을 구현
3) BigArray_Algorithms.h 파일에는 BigArray_Algorithms.cpp 파일에 포함된 함수들의 함수 원형과 필요한 기호 상수 (symbolic constant), 전처리기 지시자 등을 포함할 것
4) RAND_MAX (32767) 보다 더 큰 값의 size와 base가 주어질 때 난수 (random number) 값의 크기가 base + (0 ~ size-1)인 중복되지 않는 난수들을 생성하여 배열에 저장하여 주는 함수 genBigRandArray(int *bigRandArray, int size, int base)를 구현하라. base값은 big rand 난수들의 최소 크기를 지정하며, 생성된 난수의 최대크기는 base + size –1의 값이 된다. genBigRandArray() 함수 호출에서 전달되는 bigRandArray는 동적으로 할당된 메모리 블록의 주소가 포인터 자료형으로 전달되며, 정수 (integer)가 size개 저장될 수 있는 공간이며, 배열로 사용될 수 있다.
5) Big Rand 난수 배열의 첫 부분과 마지막 부분의 샘플만을 출력하기 위하여 void printBigArraySample(int *bigArray, int size, int items_per_line, int num_sample_lines) 함수를 작성하라. 이 함수는 주어진 bigArray[] 배열의 첫 부분에서 한 줄에 items_per_line 개씩 num_sample_line 줄을 출력하고, 배열의 마지막 부분에서 한 줄에 items_per_line 개씩 num_sample_line 줄을 출력하며, 중간 부분은 “. . . . .” 표시를 출력한다.
6) 정수배열을 선택 정렬 (selection sorting) 알고리즘으로 정렬하는 함수 selectionSort(int *bigRandArray, int size)함수를 구현하라.
7) genBigRandArray(), printBigArraySample(), selectionSort() 함수들은 BigArray_Algorithms.cpp 파일에 구현하고, 그 함수 원형은 BigArray_Algorithms.h에 포함시킬 것.
8) main() 함수에서는 50000 ~ 100000 범위의 배열 크기 size를 입력받아 동적 배열을 생성하고, genBigRandArray() 함수를 호출하여 중복되지 않은 난수 배열을 생성한 후, printBigArraySample() 함수를 사용하여 첫부분과 끝 부분의 샘플을 출력. selectionSort() 함수를 호출하여 오름차순으로 정렬한 후, printBigArraySample() 함수를 사용하여 정렬된 배열 내용을 출력.

```
/* 주석문 */
. . .  // 필요한 전처리기 포함
int main()
{
        int* int_array;
        int array_size;

        srand(0);
        printf("Input array size : ");
        scanf("%d", &array_size);
        int_array = (int*)calloc(array_size, sizeof(int));
        genBigRandArray(int_array, array_size, -array_size / 2);
        printf("\nGenerated big random array ...\n");
        printBigArraySample(int_array, array_size, 10, 3);

        printf("\nSorting big random array ...\n");
        selectionSort(int_array, array_size);
        printf("\n... After sorting ...\n");
        printBigArraySample(int_array, array_size, 10, 3);
        return 0;
}
```

9) 실행결과 (예시)

```
Input array size : 100000

Generated big random array ...
    2903   -20779    22437   -13395     6212   -30796   -28663    22337   -13955   -29624
   23459   -33054   -49701    24848   -48922     1118    -6293   -20187    47865     9075
   -5778   -20849    41080    43927   -17030   -33524    21777     7633   -46152   -48528
  . . . . .
   13567    15208    39079    34208   -34220   -29161    -8116   -47442   -45313    -5725
  -11424   -11732    20043   -28074    25263    29430    20719    32413    15825    32536
  -27946   -48754   -12846    13028   -47778   -34954    14022   -39836    21878    16104

Sorting big random array ...

... After sorting ...
  -50000   -49999   -49998   -49997   -49996   -49995   -49994   -49993   -49992   -49991
  -49990   -49989   -49988   -49987   -49986   -49985   -49984   -49983   -49982   -49981
  -49980   -49979   -49978   -49977   -49976   -49975   -49974   -49973   -49972   -49971
  . . . . .
   49970    49971    49972    49973    49974    49975    49976    49977    49978    49979
   49980    49981    49982    49983    49984    49985    49986    49987    49988    49989
   49990    49991    49992    49993    49994    49995    49996    49997    49998    49999
```