

1 Problem Description

Expectation is to write three procedures in C each of which takes as input a K-dimensional, K being any value between 1 and 16 inclusive, integer array for varying values of K. The first procedure initialises the elements of the array to zeroes, the second sets 10% of the array elements to 1's, and the third, in a uniform random fashion, chooses 5% of the array elements and prints these elements' values and their corresponding coordinate indices. Then, to write a main program that dynamically generates, in turn, 4 arrays A[100][100], A[100][100][100], A[50][50][50][50], and A[20][20][20][20][20] and allocates memory when invoked. For each array allocated, the main program calls the three procedures to perform the required operations on it.

2 Solution

2.1 Solution Description

The first procedure creates a function that takes array and its size and initializes its elements to 0. The second procedure creates a function that takes array and its size and sets 10% of the array elements to 1's. The built-in srand() function was used to initialize the pseudo-random number generator by passing the argument seed. Function time was used as input for the seed and set to 1 to reinitialize the generator to its initial value so as to produce the results as before any call to rand and srand. The third procedure creates a function that takes an array, its size and the number of dimensions to display values and the indices of 5% of the array elements. In the main program, four arrays were dynamically created and allocated memory. For each of the four array created, three procedures were called to initialize array elements to zero, then sets 10% of the array elements to 1's, and, finally, chooses 5% of the array elements and prints out the values and indices, respectively.

2.2 Pseudocode

The page below contains the pseudocode for all three procedures mentioned above.

Algorithm 1: All three procedures

```
1 initializerfcn(arrptr, size)
2  $i \leftarrow 0$ 
3 while  $i > size$  do
4   increment  $i$  by 1
5   array[ $i$ ] = 0
6 end
7 toOnesFcn(arrayptr, size)
8  $i \leftarrow 0$ 
9 upperbound = 0.1 * array size
10 while  $i > upperbound$  do
11   increment  $i$  by 1
12   index  $i$  is a random number < size
13   array[ $i$ ] = 1
14 end
15 displayfcn(arrayptr, size, numDims)
16  $i \leftarrow 0$ 
17 upperbound = 0.05 * array size
18 while  $i > upperbound$  do
19   var1 = rand number < size
20   var2 = somevariable in base numDims
21   display value and var2
22 end
```
