```
// VERSION “0”  -  static array + static size
// using TWO constants
#include <stdio.h>
#include <stdlib.h>
#define  SIZE  5
float  static_array [SIZE];


int main()
{
    // [START] the array is created automatically
    // (no separate code creating the array)


    // filling the array with data
    int i;
    for(i=0; i<SIZE; i++)
              scanf("%f", &static_array[i] );


    // sample statistical calculations
    float sum=0;
    for(i=0; i<SIZE; i++)
              sum += static_array[i];
    float average = sum/SIZE;
    printf("The average value: %.2f", average);


    // no separate code to delete the array
    // [FINISH] the array is deleted automatically

    return 0;
}
```

```
// VERSION “1” -  static array + logical
allocation
// using TWO constants + ONE variable
#include <stdio.h>
#include <stdlib.h>
#define  MAX_SIZE  100
float  static_array [MAX_SIZE];
Int  logical_size = 0;

int main()
{
    // [START] the array is created automatically
    // (no separate code creating the array)

    // "logical" allocation of the array
    printf("How many elements: ");
    scanf("%d", &logical_size );
    if( logical_size > MAX_SIZE )
        logical_size = MAX_SIZE;

    // filling the array with data
    int i;
    for(i=0; i<logical_size; i++)
        scanf("%f", &static_array[i] );

    // sample statistical calculations
    float sum=0;
    for(i=0; i<logical_size; i++)
        sum += static_array[i];
    float average = sum/logical_size;
    printf("The average value: %.2f", average);

    // "logical" deallocation of the array
    logical_size = 0;

    // no separate code to delete the array
    // [FINISH] the array is deleted automatically

    return 0;
}
```

```
// VERSION “2” – dynamic array
// using TWO variables  (array pointer + size)
#include <stdio.h>
#include <stdlib.h>
float*  dynamic_array = NULL;
int  dynamic_size = 0;


int main()
{
    // [START] the array is NOT created automatic.
    // it is NOT existing yet

    //"physical" allocation of the array (by develop.)
    printf("How many elements: ");
    scanf("%d", &dynamic_size );
    dynamic_array = (float*) calloc( dynamic_size,
                                         sizeof(float));
    if( dynamic_array == NULL )
        dynamic_size = 0;

    // filling the array with data
    int i;
    for(i=0; i<dynamic_size; i++)
              scanf("%f", &dynamic_array[i] );

    // sample statistical calculations
    float sum=0;
    for(i=0; i<dynamic_size; i++)
              sum += dynamic_array[i];
    float average = sum/dynamic_size;
    printf("The average value: %.2f", average);

    // "physical" deallocation (by developer)
    if( dynamic_array != NULL )
        {
            free( dynamic_array );
            dynamic_array = NULL;
        }
    dynamic_size = 0;

    // [FINISH] the array is finally NOT existing here
    return 0;
}
```

```
// VERSION “3”  -  using THREE variables
// (array pointer + allocated_size + logical_size)
#include <stdio.h>
#include <stdlib.h>
float*  dynamic_array = NULL;
int  allocated_size = 0;
int  logical_size = 0;
#define BLOCK_SIZE 10

int main()
{
    // [START] the default array is allocated (by dev.)
    allocated_size = BLOCK_SIZE;
    dynamic_array = (float*) calloc( allocated_size,
                                          sizeof(float));
    if( dynamic_array == NULL )
        allocated_size = 0;

    // "logical" allocation (+optional reallocation)
    printf("How many elements: ");
    scanf("%d", &logical_size );
    if( logical_size <= allocated_size )
        ;    // nothing to do, the area is ready
    else
    { // the allocated area is too small, reallocation ! !
        int new_size = (logical_size / BLOCK_SIZE + 1)
                                  * BLOCK_SIZE;
        dynamic_array = (float*) realloc( dynamic_array,
                                      new_size * sizeof(float));
        if( dynamic_array == NULL )
            {
                allocated_size = 0;
                logical_size = 0;
            }
        else
            allocated_size = new_size;
    }

    // filling the array with data
    int i;
    for(i=0; i<logical_size; i++)
            scanf("%f", &dynamic_array[i] );
    // sample statistical calculations
    float sum=0;
    for(i=0; i<logical_size; i++)
            sum += dynamic_array[i];
    float average = sum/logical_size;
    printf("The average value: %.2f", average);

    // "physical" deallocation of the array (by devel.)
    if( dynamic_array != NULL )
        {
            free( dynamic_array );
            dynamic_array = NULL;
            allocated_size = 0;
        }
    logical_size = 0;

    // [FINISH] the array is finally NOT existing here
    return 0;
}
```