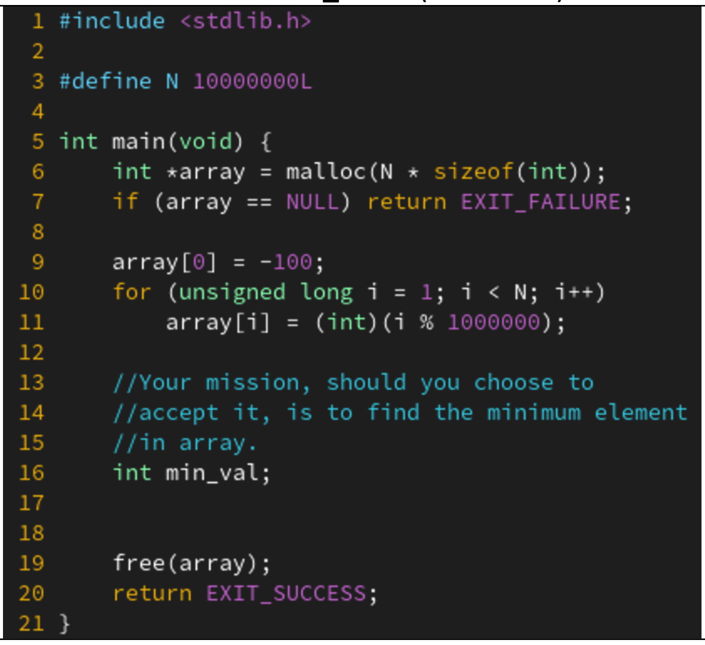
National University of Computer and Emerging Sciences

Lahore Campus

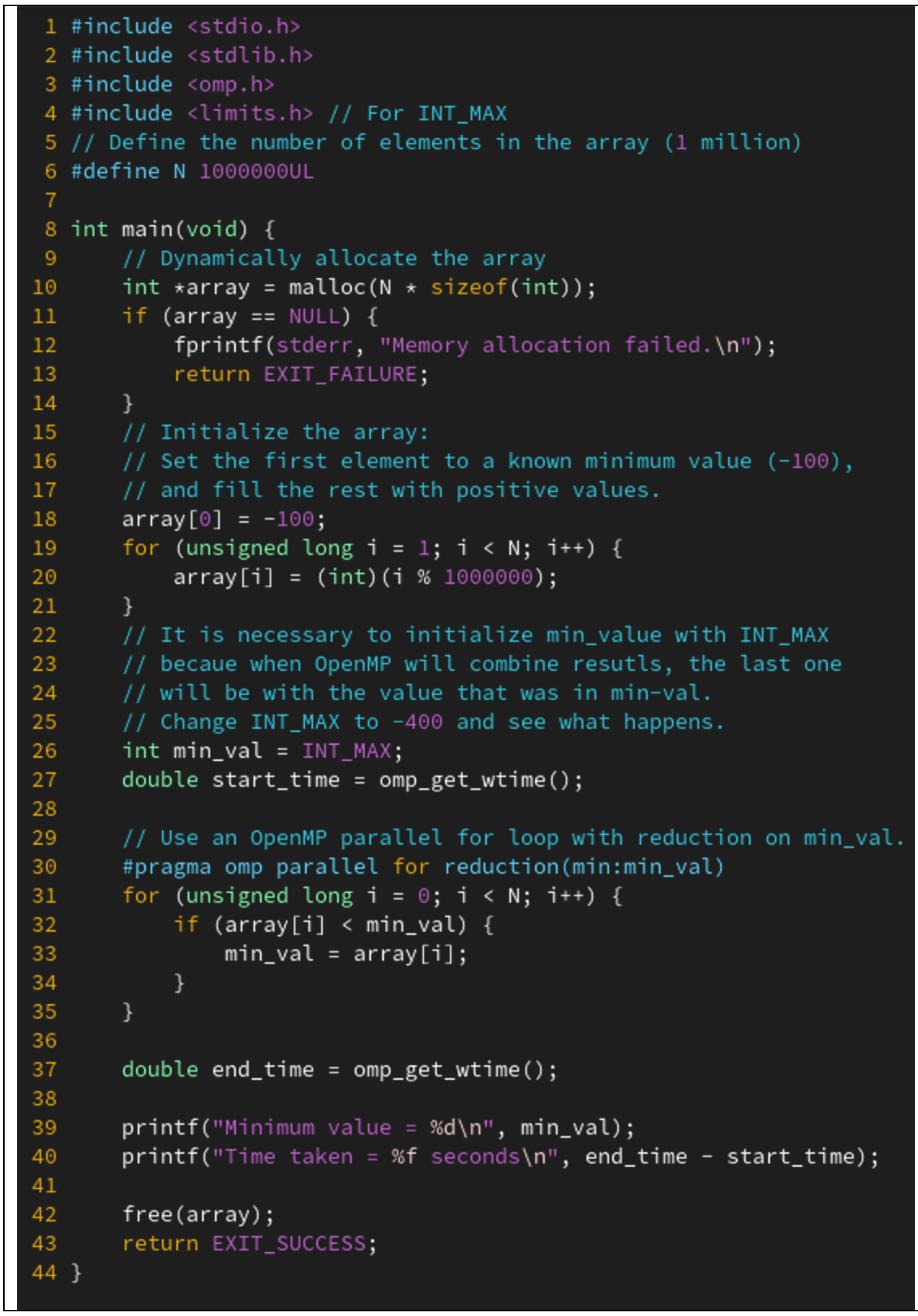
Quiz 4
Total Marks : 5
Time allowed: 5 Minute
Date: March 11th , 2025

Q1: [5 Marks] See the following code. Find the minimum in an integer array using OpenMP parallelism. We want our minimum value in **min_value** (line no 16).



```
 1 #include <stdlib.h>
 2
 3 #define N 10000000L
 4
 5 int main(void) {
 6     int *array = malloc(N * sizeof(int));
 7     if (array == NULL) return EXIT_FAILURE;
 8
 9     array[0] = -100;
10     for (unsigned long i = 1; i < N; i++)
11         array[i] = (int)(i % 1000000);
12
13     //Your mission, should you choose to
14     //accept it, is to find the minimum element
15     //in array.
16     int min_val;
17
18
19     free(array);
20     return EXIT_SUCCESS;
21 }
```

Solution:



```c
#include <stdio.h>
#include <stdlib.h>
#include <omp.h>
#include <limits.h> // For INT_MAX
// Define the number of elements in the array (1 million)
#define N 1000000UL

int main(void) {
    // Dynamically allocate the array
    int *array = malloc(N * sizeof(int));
    if (array == NULL) {
        fprintf(stderr, "Memory allocation failed.\n");
        return EXIT_FAILURE;
    }
    // Initialize the array:
    // Set the first element to a known minimum value (-100),
    // and fill the rest with positive values.
    array[0] = -100;
    for (unsigned long i = 1; i < N; i++) {
        array[i] = (int)(i % 1000000);
    }
    // It is necessary to initialize min_value with INT_MAX
    // becaue when OpenMP will combine resutls, the last one
    // will be with the value that was in min-val.
    // Change INT_MAX to -400 and see what happens.
    int min_val = INT_MAX;
    double start_time = omp_get_wtime();

    // Use an OpenMP parallel for loop with reduction on min_val.
    #pragma omp parallel for reduction(min:min_val)
    for (unsigned long i = 0; i < N; i++) {
        if (array[i] < min_val) {
            min_val = array[i];
        }
    }

    double end_time = omp_get_wtime();

    printf("Minimum value = %d\n", min_val);
    printf("Time taken = %f seconds\n", end_time - start_time);

    free(array);
    return EXIT_SUCCESS;
}
```