```c
#define NTHR 8
#define NARR (1024*1024)
int array[NARR];

typedef struct Args {
    int start; int end; int max;
} Args;

Args *
search(Args *args)
{
    args->max = array[0];
    for (int i = args->start;
         i < args->end;
         i++) {
        if (args->max < array[i])
            args->max = array[i];
    }

    return args;
}

int
main(int argc, const char *argv[])
{
    pthread_t thr[NTHR];
    Args args[NTHR];

    void *rval;

    // Fill in random values
    arc4random_buf(&array[0], sizeof(array));

    for (int i = 0; i < NTHR; i++) {
        args[i].start = NARR/NTHR*i;
        args[i].end = NARR/NTHR*(i+1);
        pthread_create(&thr[i], NULL, &search, &args[i]);
    }

    for (int i = 0; i < NTHR; i++) {
        pthread_join(thr[i], &rval);
    }

    int max = args[0].max;
    for (int i = 0; i < NTHR; i++) {
        if (max < args[i].max)
            max = args[i].max;
    }

    printf("Max: %d, Max (hex): %08x\n", max, max);

    return 0;
}
```

Question 1. What does the function do?

Question 2. If you have eight cores what is the optimal value for NTHR? Why?

Question 3. If you have 1024 cores what is the optimal value for NTHR? Why?

Question 4. What is the expected output (give a number)?