2.

#include <stdio.h>

#include <stdlib.h>

#include <pthread.h>

#include <unistd.h>

*void*\* threadFunction (*void* \**value*)

{

*int* random\_num = 0;

*int* guessed\_num = 0;

*int* counter = 0;

random\_num = rand() % 10 + 1;

while(1)

{

printf("\nGuess the number : \n");

counter++;

scanf("%d", &guessed\_num);

usleep (1000);

if (guessed\_num == random\_num)

{

printf("You guessed correctly in %d tries! Congratulations!\n", counter);

break;

}

if (guessed\_num < random\_num)

printf("Your guess is too low. Guess again. ");

if (guessed\_num > random\_num)

printf("Your guess is too high. Guess again. ");

}

//printf("random is %d",random\_num);

}

*int*

main ()

{

*pthread\_t* thread;

if (pthread\_create (&thread, NULL, &threadFunction, (*void*\*)NULL) != 0)

{

printf ("Failed to create the thread\n");

return 1;

}

*void* \*result;

pthread\_join (thread, &result);

*int* \*z = (*int*\*) result;

//printf ("Final retrun : z=%d\n", \*z);

return EXIT\_SUCCESS;

}

Binary\_search :

#include <stdio.h>

#include <stdbool.h>

#include <stdlib.h>

#include <pthread.h>

#include <unistd.h>

#define MAX 16

#define MAX\_THREAD 4

*int* a[] = { 1, 5, 7, 10, 12, 14, 15, 18, 20, 22, 25, 27, 30, 64, 110, 220 };

*int* key = 100;

*bool* found = false;

*int* part = 0;

*void*\* binary\_search(*void*\* *arg*)

{

// Each thread checks 1/4 of the array for the key

*int* thread\_part = part++;

*int* mid;

*int* low = thread\_part \* (MAX / 4);

*int* high = (thread\_part + 1) \* (MAX / 4);

while (low < high && !found) {

mid = (high - low) / 2 + low;

if (a[mid] == key) {

found = true;

break;

}

else if (a[mid] > key)

high = mid - 1;

else

low = mid + 1;

}

}

*int* main()

{

*pthread\_t* threads[MAX\_THREAD];

/\* printf("Enter number of elements in array\n");

scanf("%d", &n);

printf("Enter %d elements\n", n);

for (int c = 0; c < n; c++)

scanf("%d", &a[c]);

printf("Enter key to search : ");

scanf("%d",key);\*/

for (*int* i = 0; i < MAX\_THREAD; i++)

pthread\_create(&threads[i], NULL, binary\_search, (*void*\*)NULL);

for (*int* i = 0; i < MAX\_THREAD; i++)

pthread\_join(threads[i], NULL);

if (found)

printf("%d found is array\n",key);

else

printf("%d not found is array\n",key);

return 0;

}