# 1.MAXIMUM SUBARRAY SUM:

#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>

unsigned long max\_sum(

unsigned long \*partial\_sums,

unsigned long \*dest,

unsigned length,

unsigned long target

) {

if (length < 2)

return (\*dest = \*partial\_sums);

unsigned

lower\_len = length >> 1,

upper\_len = length - lower\_len;

unsigned long

lower[lower\_len],

upper[upper\_len],

max\_left = max\_sum(partial\_sums, lower, lower\_len, target),

max\_right = max\_sum(&partial\_sums[lower\_len], upper, upper\_len, target),

max = max\_left > max\_right ? max\_left : max\_right;

unsigned lesser = 0, greater = 0;

while (lesser < lower\_len && greater < upper\_len)

if (upper[greater] < lower[lesser]) {

\*dest = upper[greater++];

unsigned long other\_max = (\*dest++ - lower[lesser]) + target;

if (other\_max > max)

max = other\_max;

} else

\*dest++ = lower[lesser++];

memcpy(dest, &lower[lesser], (lower\_len - lesser) \* sizeof(\*dest));

memcpy(&dest[lower\_len - lesser], &upper[greater], (upper\_len - greater) \* sizeof(\*dest));

return max;

}

int main() {

unsigned length;

unsigned long test\_case, limit, index;

static unsigned long

items[100000],

buffer[100001] = {[0] = 0},

\*partial\_sums = &buffer[1];

for (scanf("%lu", &test\_case); test\_case--; ) {

scanf("%u", &length);

scanf("%lu", &limit);

for (index = 0; index < length; index++) {

scanf("%lu", &items[index]);

items[index] %= limit;

partial\_sums[index] = (items[index] + partial\_sums[index - 1ULL]) % limit;

}

static unsigned long ordered[100000];

unsigned long max = max\_sum(partial\_sums, ordered, length, limit);

printf("%lu\n", max);

}

return 0;

}