682. Sum on a segment

Given array of integers a_1, ..., a_n. For given indexes l and r find

$$S_{l,r} = a_l + a_{l+1} + .. + a_r$$

Input. The first line contains the amount of numbers n ($1 \le n \le 10^6$). Second line contains the numbers a_i ($1 \le a_i \le 1000$) space separated. Third line contains the number of queries m ($1 \le m \le 10^6$). Each of the next line contains the query l_i and r_i ($1 \le l_i \le r_i \le n$).

Output. Print in separate m lines the numbers $S_{li..ri}$.

Sample input	**Sample output**
5 1 2 3 4 5 5 1 5 2 3 3 4 2 5 1 4	15 5 7 14 10

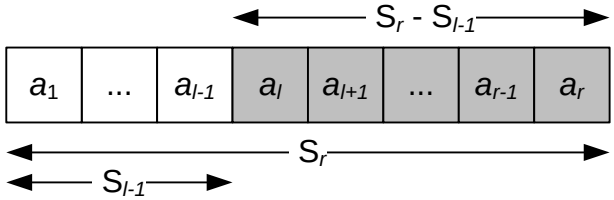
Algorithm analysis

Each query $S_{l,r}$ can be performed using a loop with complexity O(n). We have $m \le 10^6$ queries, the length of array is $n \le 10^6$. If the query is run is linear time, we need no more than $n * m \le 10^{12}$ operations, it will give us Time Limit.

Consider the partial sums of array a:

$s_1 = a_1$;
$s_2 = a_1 + a_2$;
$s_3 = a_1 + a_2 + a_3$;
. . .
$s_n = a_1 + a_2 + a_3 + \ldots + a_n$;

Its possible to find the values s_1, s_2, …, s_n in linear array s in O(n) time.



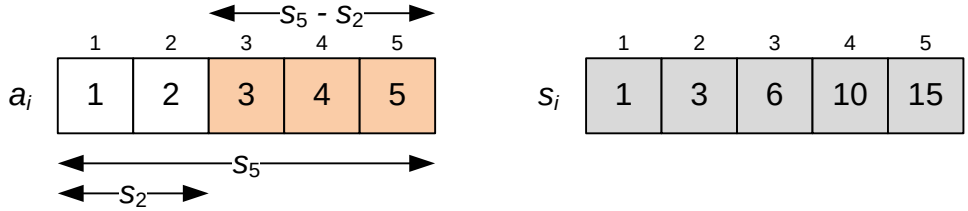
Further note that

$$S_{l,r} = a_l + a_{l+1} + \ldots + a_r =$$
$$= (a_1 + \ldots + a_l + a_{l+1} + \ldots + a_r) - (a_1 + \ldots + a_{l-1}) = s_r - s_{l-1}$$

So we can answer the query $S_{l,r}$ in constant time.

Example

Let's find the sum $a_3 + a_4 + a_5$ for the input example.



We have: $a_3 + a_4 + a_5 = s_5 - s_2 = 15 - 3 = 12$.

4739. Sieve of Eratosthenes

Given the value of *a* and *b*, print all primes in the interval from *a* to *b* inclusively.

Input. Two integers *a* and *b* ($1 \le a \le b \le 100000$).

Output. Print in one line all prime numbers in the interval from *a* to *b* inclusively.

Sample input	**Sample output**
1 10	2 3 5 7

Algorithm analysis

Using the Eratosthenes sieve algorithm, fill the primes array, where

- primes[*i*] = 1, if *i* is prime;
- primes[*i*] = 0, if *i* is composite;

Print all numbers *i* in the interval from *a* to *b*, for which primes[*i*] = 1.

Example

The filled array **primes** has the form:

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
primes[i]	0	1	1	0	1	0	1	0	0	0	1	0	1	0	0

The prime numbers in the interval [1; 10] are 2, 3, 5, 7.

Algorithm realization

Declare the working array primes.

```
#define MAX 100001
int primes[MAX];
```

Function ***gen_primes*** runs the sieve of Eratosthenes and fills the array primes. The numbers 0 and 1 are not prime.

```
void gen_primes(void)
{
```

```
  int i, j;
  for (i = 0; i < MAX; i++) primes[i] = 1;
  primes[0] = primes[1] = 0;
  for (i = 2; i * i < MAX; i++)
    if (primes[i])
      for (j = i * i; j < MAX; j += i) primes[j] = 0;
}
```

The main part of the program. Fill the primes array.

```
gen_primes();
```

Read the input data.

```
scanf("%d %d", &a, &b);
```

Print all primes in the interval from *a* to *b*.

```
for (i = a; i <= b; i++)
  if (primes[i]) printf("%d ", i);
printf("\n");
```