Homework 4

Amin Parchami, Kamran Akbar

After having a week off, Alice has just got a new homework and wants to have a fresh start, so she asks you to help her get the full mark.

Your task is to complete the body of the **`find_primes`** function used below. The function must be written **in assembly**.

file: main.c

```c
#include <stdio.h>

void read_array(int a[],int n);
void print_array(int a[],int n);
// Assume that the read_array and print_array
// functions have already been implemented.

int find_primes(int inputs[],int primes[],int n);

int main(){
    int n;
    scanf("%d",&n);
    int inputs[n];
    read_array(inputs,n);
    int primes[n];

    //TODO: Implement find_primes in assembly.
    int count = find_primes(inputs,primes,n);

    print_array(primes,count);
    return 0;
}
```

You must write your assembly code in two separate files:

- **find.asm**
- **check.asm**

In **find.asm** write the body of the `find_primes` function. The first argument is the input array which contains positive integers. The second argument is the output array. The last argument **n** is the size of the input array, that is the number of integers in the input array. The function must find the prime numbers from the input array and store them in the output array. Besides, it must return the number of primes found (size of the output array) as the return value.

Obviously, the function `find_primes` needs to check if an integer is prime. To do so, you must write a separate function called **is_prime** in assembly. This function receives an integer as argument and returns 1 or 0 depending on whether or not the input argument is prime. This assembly function must be written in a separate file called **"check.asm".**

You must also create a **Makefile.** It must assemble, compile and link the input files and create a 32-bit executable named **run.out.**

Input:

The first line of input contains a positive integer between 1 and 100 (call it **n**). This is the number of positive integers to be checked (the size of the input array). The second line contains **n** positive integers.

Output:

The output contains the list of the prime numbers from the input numbers in the same order.

Notice that the C code in main.c takes care of getting the input and producing the output.

Note that:

- You must observe the default C calling conventions **for both functions `find_primes`** and **`is_prime`**.
- Your code **must** work with the **main.c** file provided with the homework. Do not change the C file.
- You must not print extra outputs. Reading and printing the numbers is done by the given C file and you **must not** read or print any numbers in your assembly files. If you do so (for the purpose of debugging, etc.) delete them before submission.
- Your code must work without the asm_io.asm, driver.c and other function provided by the book. If you link your code against **asm_io.o** for debugging, make sure that remove this dependency before the submission.
- Your code must work under a Linux platform.

Please upload only the following files on courses.kntu.ac.ir:

- find.asm
- check.asm
- Makefile

Your code will be checked for similarity. In the case of cheating, the student will receive a **negative** point. It is **your responsibility** to protect your code.

Example :

Input:

7

10 2 4 1 11 9 17

Output:

```
2 11 17
```

Solution:

2, 11, and 17 are the prime numbers from the given series.

Extra Credit!

Write the 64-bit version of the above (beside the 32-bit version). You must observe all the 64-bit (linux) calling conventions for both functions. The 64-bit version must be submitted in a separate directory.