

```
#define megisto_plithos ???
```

```
??? displaySet(???, ???, ???)
```

```
??? createPrimeSet(???, ???, ???)
```

```
??? isPrime(???)
```

```
main(){
```

```
    Δήλωσε το σύνολο typos_synolou Synolo
```

```
    Επανάλαβε {
```

```
        Διάβασε το first
```

```
    } όσο το first δεν ανήκει στο [2...200]
```

```
    Επανάλαβε {
```

```
        Διάβασε το last
```

```
    } όσο το last δεν ανήκει [2...200] ή το last <= first
```

```
    //Δημιούργησε το σύνολο των πρώτων αριθμών
```

```
    Κάλυψε τη συνάρτηση createPrimeSet (???, ???, ???)
```

```
    //Εμφάνισε το σύνολο των πρώτων αριθμών
```

```
    Κάλυψε τη συνάρτηση displaySet (???, ???, ???)
```

```
}
```

```
void createPrimeSet(typos_synolou s, stoixeio_synolou first, stoixeio_synolou last){
```

```
    Δημιούργησε το σύνολο s
```

```
    Με επαναληπτική δομή από ??? έως ??? με βήμα 1
```

```
        Έλεγχος αν ο τρέχων αριθμός είναι πρώτος με κλήση της isPrime(τρέχων αριθμός)
```

```
        //Εισήγαγε στο σύνολο s τον τρέχοντα αριθμό i
```

```
        Κάλυψε τη συνάρτηση Eisagogi(???, ???);
```

```
}
```

```
void displaySet(typos_synolou set, stoixeio_synolou first, stoixeio_synolou last){
```

```
    Με επαναληπτική δομή διέτρεξε τα στοιχεία του συνόλου από ??? έως ??? με βήμα 1
```

```
    Έλεγχος αν ο τρέχων αριθμός ανήκει στο σύνολο
```

```
    Τύπωσε τον τρέχοντα αριθμό
```

```
    Μετακίνησε το κέρσορα στην επόμενη γραμμή
```

```
}
```

boolean isPrime(stoixeio_synolou n)

{
stoixeio_synolou i, limit;

Αν ο n είναι το 2 επέστρεψε αληθές

Αν ο n είναι άρτιος επέστρεψε ψευδές

//Αρκεί να ελέγξουμε τη διαίρεση του n με τους αριθμούς \leq της τετραγωνικής του ρίζας

limit=???

Με επαναληπτική δομή από ??? μέχρι ??? με βήμα ???

Αν ο n διαιρείται ακριβώς επέστρεψε ψευδές

Επέστρεψε αληθές

}