

Esercizi

Andrea Marin

Università Ca' Foscari Venezia
Laurea in Informatica
Corso di Programmazione part-time

a.a. 2011/2012

Test di primalità

Definizione (Numeri primi)

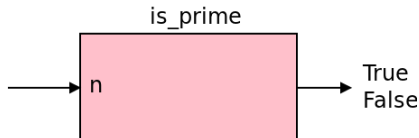
Un numero naturale è primo se è maggiore di 1 e ammette come divisori solo se stesso ed 1

- ▶ La ricerca dei numeri primi è un tema affascinante e ancora molto attuale
- ▶ Nelle prossime diapositive affronteremo il problema della ricerca dei numeri primi da un punto di vista diattico
 - ▶ Lo scopo è scrivi buoni programmi prima ancora di scriverli veloci!



Impostare il problema

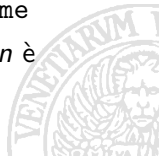
- ▶ Faremo uso del sottoprogramma `is_prime`:



- ▶ Prototipo della funzione:

```
int is_prime(int n);
```

- ▶ Daremo diverse implementazioni della funzione `is_prime`
- ▶ La funzione `is_prime` restituisce la codifica di *true* se n è primo, 0 altrimenti



Contare i divisori

- ▶ Idea risolutiva:
 - ▶ Conto i divisori compresi tra 1 e n inclusi
 - ▶ Se sono esattamente 2 allora il numero è primo, altrimenti non lo è
- ▶ Ciclo:
 - ▶ Per provare se un numero i compreso fra 1 e n è un divisore dobbiamo impostare un ciclo
 - ▶ Sappiamo quante iterazioni deve fare?
 - ▶ Sì, sono esattamente n
 - ▶ Quindi useremo il ciclo `for`



Soluzione 1

```
int is_prime(int n) {  
    int i, divisori;  
    divisori = 0;  
    for (i=1; i<=n; i=i+1) {  
        if (n%i == 0) {  
            divisori = divisori + 1;  
        }  
    }  
    return (divisori == 2);  
}
```



Altro punto di vista

- ▶ Cerco di ricondurmi ad un pattern noto
- ▶ $n > 1$ è *primo* se tutti i numeri tra 2 e $n - 1$ sono coprimi con n
- ▶ Devo testare una proprietà su un insieme
 - ▶ Assumo che sia vera
 - ▶ e cerco l'elemento che la falsifica (un divisore di n)
- ▶ Che ciclo usare?
 - ▶ Ancora il ciclo `for`



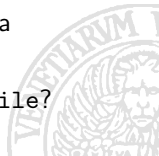
Soluzione 2

```
int is_prime(int n) {
    int i, primo;
    if (n > 1) {
        primo = 1;
    }
    else {
        primo = 0;
    }
    for (i=2; i < n; i=i+1) {
        if (n%i == 0) {
            primo = 0;
        }
    }
    return primo;
}
```



Interrompere il ciclo

- ▶ Osserviamo che non appena è stato trovato un divisore è inutile proseguire con la ricerca. . .
- ▶ Il ciclo che cerca i divisori può essere interrotto non appena un divisore è stato trovato
- ▶ Il C mette a disposizione l'istruzione `break` che interrompe il ciclo che governa il blocco di cui fa parte
- ▶ La prossima slide mostra l'uso del `break` che **però è sconsigliabile!**
 - ▶ In questo corso eviteremo di usare il `break` come regola
- ▶ Adesso so quante iterazioni devo compiere?
 - ▶ No, quindi al posto del `break` perchè non passare al `while`?



Soluzione col break (non usare)

```
int is_prime(int n) {
    int i, primo;
    if (n>1) {
        primo = 1;
    }
    else {
        primo = 0;
    }
    for (i=2; i<n; i=i+1) {
        if (n%i == 0) {
            primo = 0;
            break;
        }
    }
    return primo;
}
```



Soluzione col return (non usare)

```
int is_prime(int n) {  
    int i, primo;  
    for (i=2; i<n; i=i+1) {  
        if (n%i == 0) {  
            return 0  
        }  
    }  
    return (n!=1);  
}
```



Soluzione con il while

```
int is_prime(int n) {
    int i, primo;
    primo = (n>1) ? 1 : 0;
    i = 2;
    while ( (i < n) && primo ) {
        if (n%i == 0) {
            primo = 0;
        }
        i = i+1;
    }
    return primo;
}
```



While vs. For

- ▶ Quando useresti il while e quanto il for?
 - ▶ Scrivere un programma che usando la `is_prime` stampi tutti i numeri primi tra 2 e 100
 - ▶ Scrivere un programma che usando la `is_prime` stampi i primi 20 numeri primi



Esercizi

1. Scrivere la funzione `count_prime` che acquisito un intero n , restituisca il numero di numeri primi $\leq n$;
2. Scrivere la funzione `next_prime` che acquisito un intero n , restituisca il più piccolo numero primo m strettamente maggiore di n
 - ▶ Esempio il primo successivo a 14 è il 17. Il successivo del 17 è il 19.
3. Scrivere un programma che legga un numero naturale da standard input e ne stampi la fattorizzazione in numeri primi

