Wrap-up

Informatica di base – a.a. 2020/2021

Silvio Peroni
0000-0003-0530-4305

Dipartimento di Filologia Classica e Italianistica, Università di Bologna, Bologna, Italia
silvio.peroni@unibo.it – @essepuntato – https://www.unibo.it/sitoweb/silvio.peroni/

DIPARTIMENTO DI FILOLOGIA CLASSICA E ITALIANISTICA

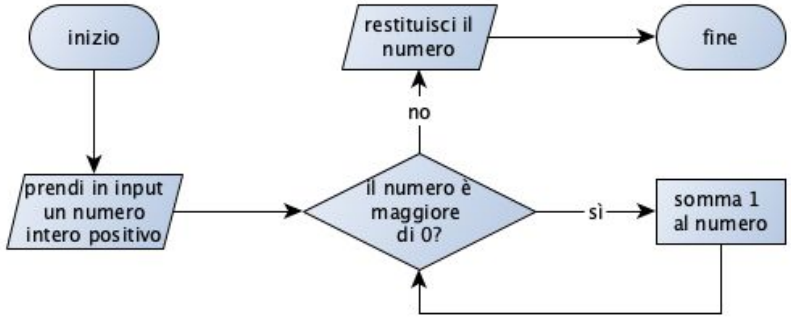
Algoritmi che non terminano e Turing

I problemi aperti della matematica

23 problemi aperti della matematica proposti da David Hilbert 1900

Includono (indirettamente) il **problema della terminazione**: capire se fosse possibile sviluppare un algoritmo che fosse in grado di rispondere se **un altro** algoritmo, specificato come input, terminasse la sua esecuzione o no

È possibile sviluppare un algoritmo che non termina mai la sua esecuzione?



Approssimazione della soluzione di Turing

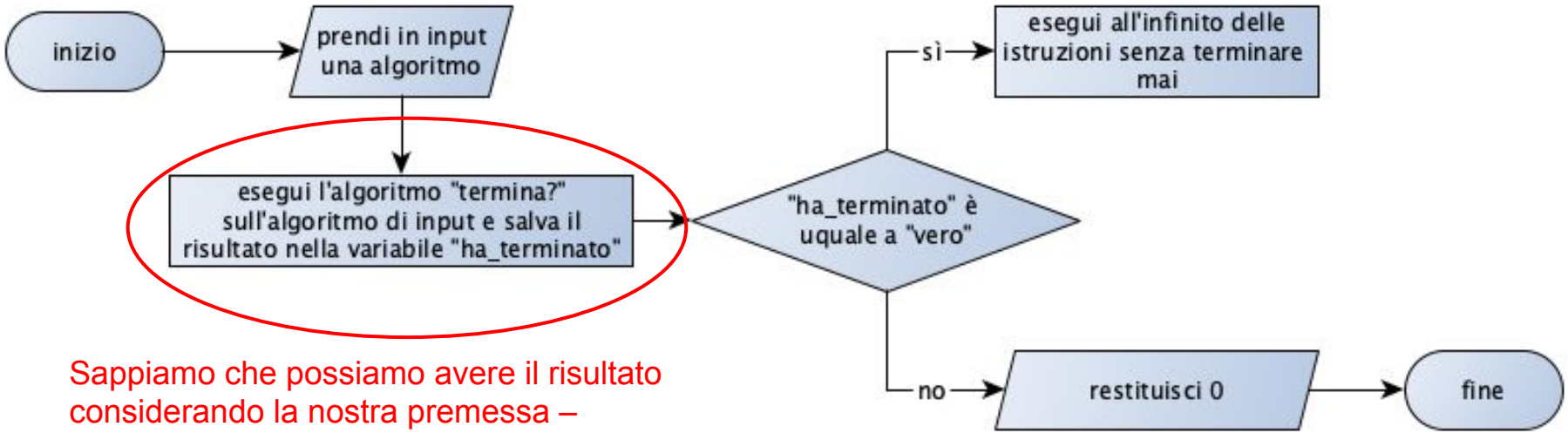
Premessa: supponiamo **sia possibile sviluppare** l'algoritmo "termina?", che prende in **input un certo algoritmo** e restituisce **"vero"** nel caso in cui l'algoritmo specificato come **input termina**, mentre restituisce **"falso" in caso contrario**

NB: è soltanto un algoritmo ipotetico, stiamo supponendo che possiamo svilupparlo in qualche modo, senza mostrare come farlo davvero

Usiamo questo algoritmo per crearne un altro

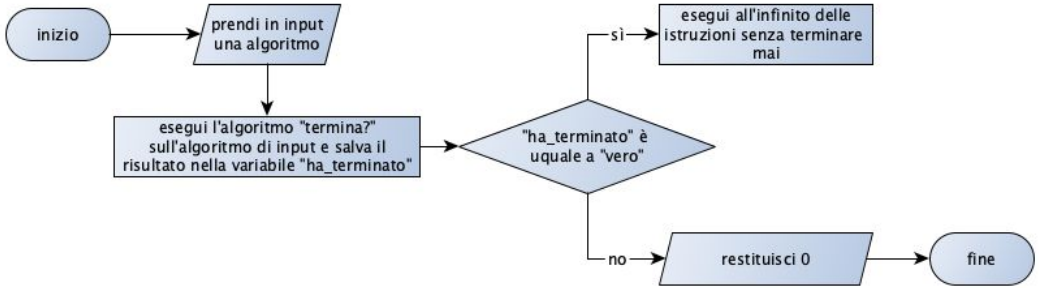
Nuovo algoritmo

Nuovo algoritmo: prendi in input un algoritmo e restituisci 0 se l'algoritmo in input termina, mentre non terminare in caso contrario



Sappiamo che possiamo avere il risultato considerando la nostra premessa – ovvero che "ha_terminato" esiste

E se usiamo il nuovo algoritmo come suo input?



Caso 1: se l'algoritmo "termina?" afferma che il nuovo algoritmo **termina**, conseguentemente (per come è definito) il nuovo algoritmo **non termina** l'esecuzione

Caso 2: se l'algoritmo "termina?" afferma che il nuovo algoritmo **non termina**, e conseguentemente (per come è definito) il nuovo algoritmo restituisce 0 e **termina** l'esecuzione

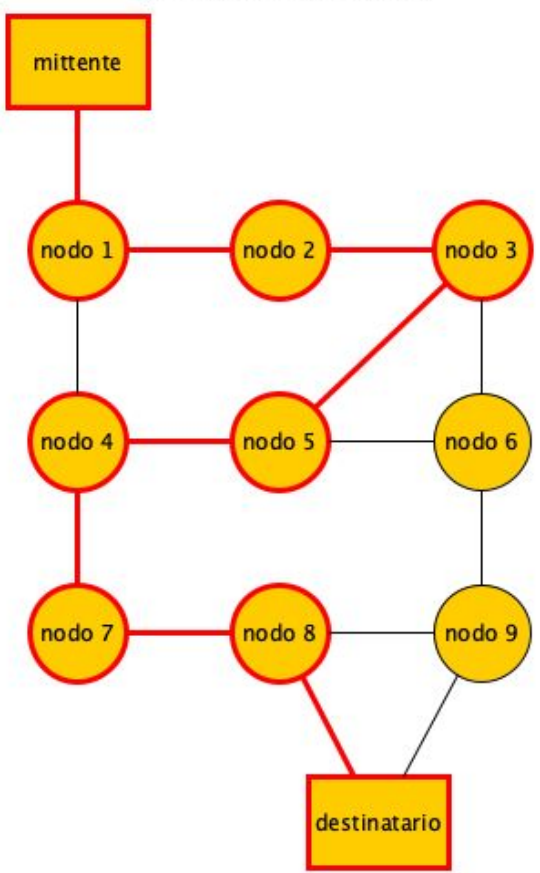
Paradosso: l'algoritmo che verifica se un altro termina **non può esistere**

Commutazione di circuito
vs
Commutazione di pacchetto

Commutazione di circuito

Viene stabilito un **circuito virtuale** unico di comunicazione tra mittente e destinatario, ovvero una sequenza finita e sequenziale di nodi riservati che permette la sola comunicazione dal mittente al destinatario

Viene usata **tutta la banda a disposizione**, che occupa in modo esclusivo i suddetti nodi fino al termine della comunicazione

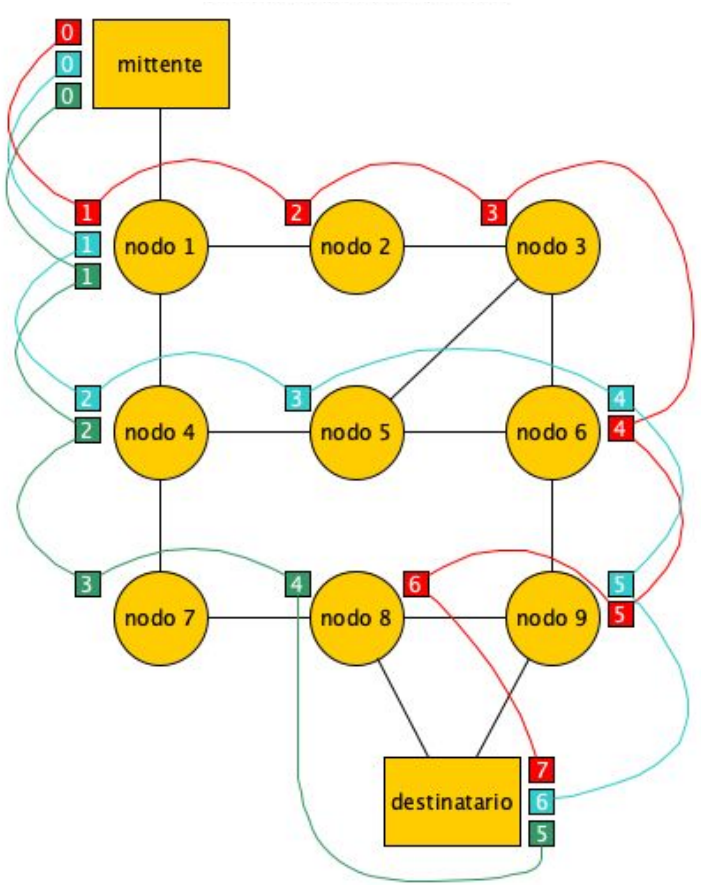


Commutazione di pacchetto

Pacchetto: **unità atomica in informazione**, formata solitamente da l'header e il payload

Dimensione **massima** in termini di informazione che possono contenere: un messaggio viene spezzato in diversi pacchetti, trasmessi sulla rete **senza occupare l'intera banda**, e instradati dal mittente al destinatario seguendo, potenzialmente, percorsi diversi

Vantaggio: tolleranza ai guasti

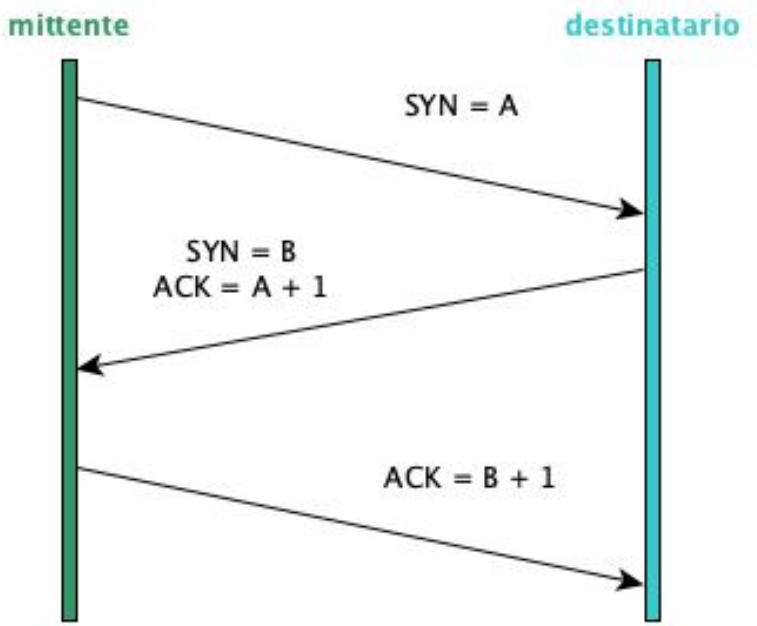


Three-way handshake

Comunicazione connessa

Il mittente e il destinatario **si mettono d'accordo** di iniziare una comunicazione in modo esplicito prima di scambiarsi i dati, e dichiarano altrettanto esplicitamente quando questa comunicazione si può ritenere conclusa

Il processo di inizio della comunicazione, è regolato dal meccanismo del **three-way handshake**, mentre quello di chiusura è il **four-way handshake**



Codifica binaria e indirizzo IP

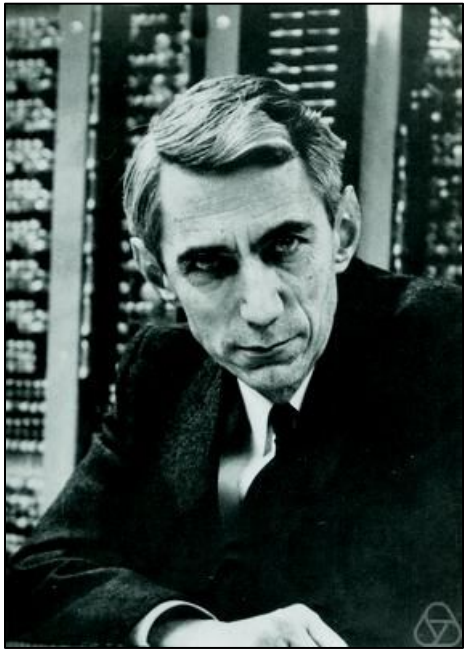
Byte e bit

Byte: unità minima di informazione occupabile su un computer, che storicamente corrisponde al numero di bit necessari per codificare un carattere sul computer

Bit (contrattura di *binary digit*): l'unità minima di informazione che si può scambiare in una comunicazione e può assumere solo uno di due valori: 0 o 1

1 byte = 8 bit

Il concetto di bit è stato usato da diversi studiosi del passato (ad esempio Babbage con le schede perforate), ma introdotto come termine formalmente da Claude Shannon (1948)



Da bit a simboli e viceversa

Ogni carattere, numero, programma, applicazione all'interno di un computer, così come un qualunque messaggio da scambiare in Internet, è **codificato** come una sequenza di bit che, in qualche modo, vengono poi **decodificati** con la sequenza di caratteri corretta dalla particolare applicazione che riceve ed interpreta quell'informazione

Per esempio, nella codifica binaria tradizionale dei numeri interi, il numero 0 è rappresentato dalla sequenza "0", il numero 1 da "1", il numero 2 da "10", il numero 3 da "11", il numero 4 da "100", e così via

Gli indirizzi IP (v4)

Sono definiti mediante l'uso di **4 byte**, uno per ogni numero

Ogni byte, ovvero 8 bit, permette di definire un **numero intero da 0 a 255**

Codifica / decodifica binaria: ognuno degli otto bit di un numero facente parte dell'indirizzo IP ha assegnato uno specifico valore – il valore 1 è associato al bit più a destra, mentre a quello successivo verso sinistra è assegnato un valore doppio rispetto al precedente (in questo caso 2), e così via

indirizzo IP	208							.	80							.	154							.	224							
bit	1	1	0	1	0	0	0	0	0	1	0	1	0	0	0	0	1	0	0	1	1	0	1	0	1	1	1	0	0	0	0	0
valore	128	64	32	16	8	4	2	1	128	64	32	16	8	4	2	1	128	64	32	16	8	4	2	1	128	64	32	16	8	4	2	1
somma	128+64		+16						64		+16						128			+16	+8		+2		128+64+32							

208 : 2 = 104 con **0** di resto
104 : 2 = 52 con **0** di resto
52 : 2 = 26 con **0** di resto
26 : 2 = 13 con **0** di resto
13 : 2 = 6 con **1** di resto
6 : 2 = 3 con **0** di resto
3 : 2 = 1 con **1** di resto
1 : 2 = 0 con **1** di resto

Idempotenza nei metodi HTTP

Classificazione metodi HTTP

Safe: viene usato solo per recuperare delle informazioni dal server web, **senza cambiarne lo stato**, ad esempio aggiungendo nuovi dati (ad esempio GET)

Idempotente: molteplici richieste effettuate allo stesso URL hanno lo **stesso effetto**, sul server web, che effettuare una **sola richiesta** (ad esempio GET e PUT)

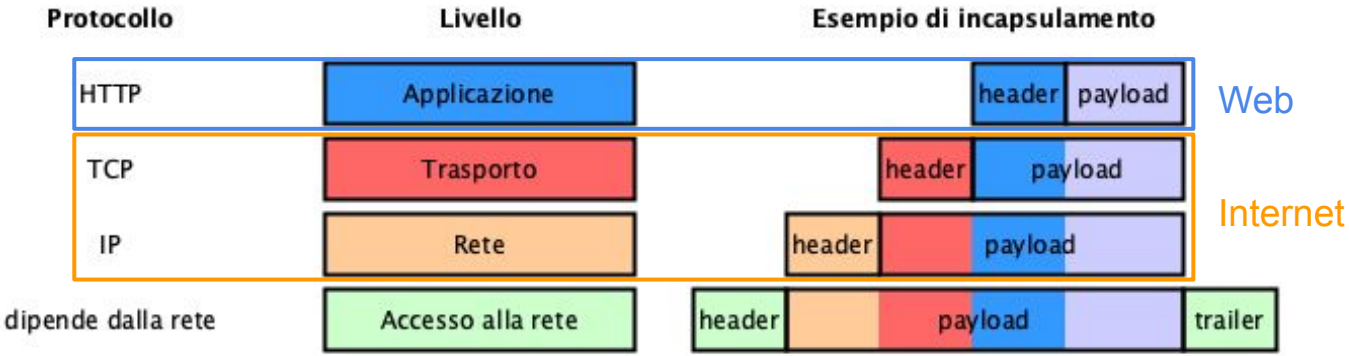
GET è preponderante nel Web, visto che viene usato per richiedere una qualsiasi risorsa, mentre gli altri metodi non sempre sono liberamente utilizzabili da tutti i client – sta sempre al server decidere quali di questi metodi possano essere usati nella comunicazione

Massima dimensione dei pacchetti

Gerarchia protocolli

Lo scambio di dati tra due computer collegati in rete è realizzata mediante l'uso dei pacchetti

Esiste una gerarchia di incapsulamento dei dati da spedire, definita dalla suite di protocolli Internet TCP/IP, organizzata in quattro livelli



Dimensione massima dei pacchetti

La scorsa lezione abbiamo accennato al fatto che un particolare messaggio debba essere **spezzato in uno o più pacchetti IP** prima che questi vengano instradati in rete

Due diversi fattori:

1. il limite dato dalla **massima quantità di dati** che ogni pacchetto IP può trasportare (che dipende dalla versione considerata del protocollo, IPv4 o IPv6)
2. il limite imposto dalla rete a cui si instradano i pacchetti, ovvero il suo **Maximum Transmission Unit (MTU)**

Com’è strutturato l’esame

Esame

La prova d’esame è composta da una prova soltanto, da sostenere a computer, plausibilmente da remoto, considerando la situazione attuale

Ogni prova è composta da 33 domande a risposta multipla, dove ogni domanda risposta correttamente vale 1 punto mentre ogni domanda risposta in modo sbagliato o non risposta vale 0 punti

A fine prova, viene restituito il risultato a video che corrisponde al voto finale

Prossimi appelli (unico esame per tutti i corsi A-D, E-M, e N-Z):
26/05/2021
16/06/2021
14/07/2021

Domande

Le domande verteranno su tutto il contenuto del corso a disposizione sulla piattaforma Virtuale – che di fatto è lo stesso dei PDF pubblicati in GitHub

Le domande sono organizzate in due “tipologie” diverse:

- 19 domande *teoriche*, che verteranno sulle conoscenze del contenuto del corso e mirano ad indagare quanto lo studente ha appreso del materiale messo a disposizione
- 14 domande di *ragionamento*, in cui si valuta la capacità di pensiero computazionale dello studente nel rispondere a particolare situazioni

Esempio domanda teorica

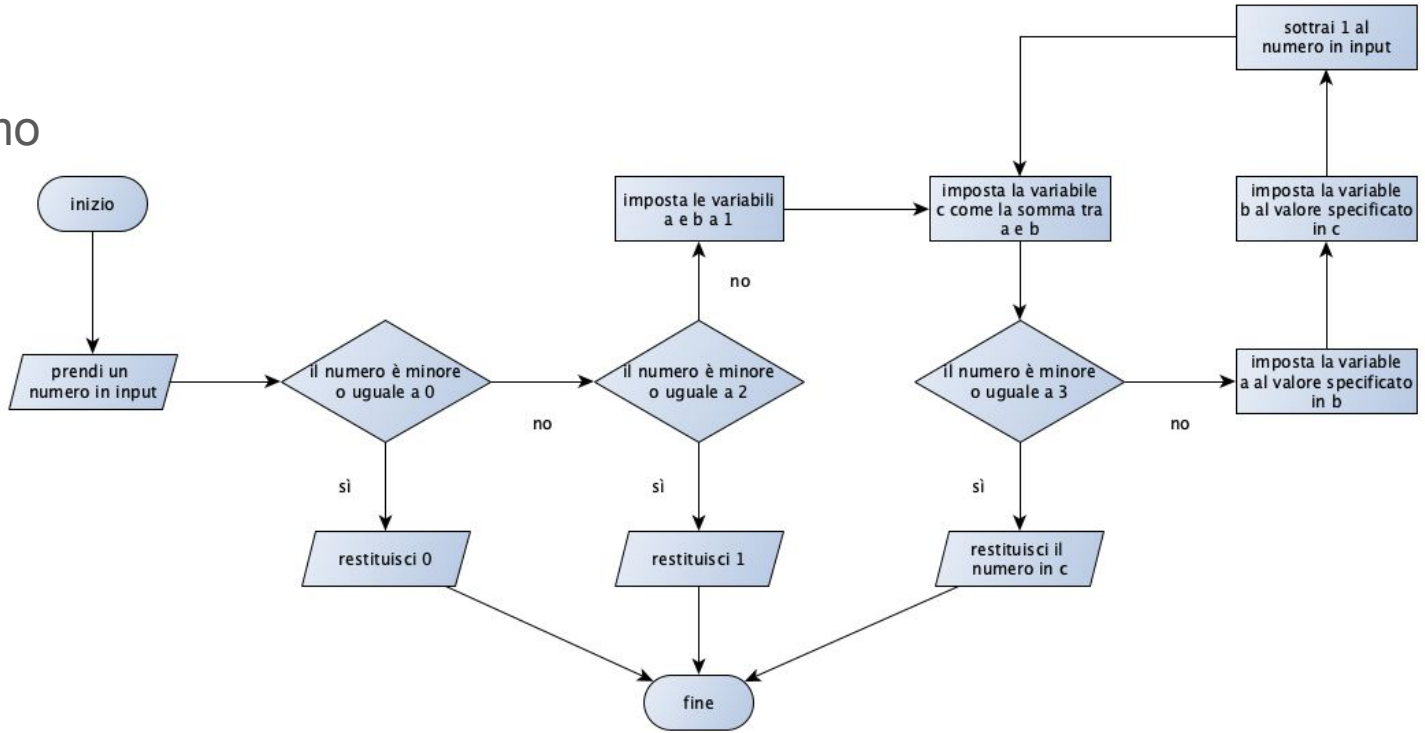
Che cos’è un programma?

- ❏ È un documento scritto in un linguaggio informale che è solitamente usato per comunicare i passi principali di un algoritmo ad un umano
- ❏ È la specifica (o implementazione), fatta da parte di un programmatore, di un certo algoritmo usando un particolare linguaggio di programmazione comprensibile da computer elettronico
- ❏ È l'astrazione di una procedura passo passo che prende qualcosa come input e produce un certo output
- ❏ È un particolare dispositivo hardware, parte di un computer elettronico, che permette di eseguire degli algoritmi a partire da specifici input

Esempio di domanda di ragionamento

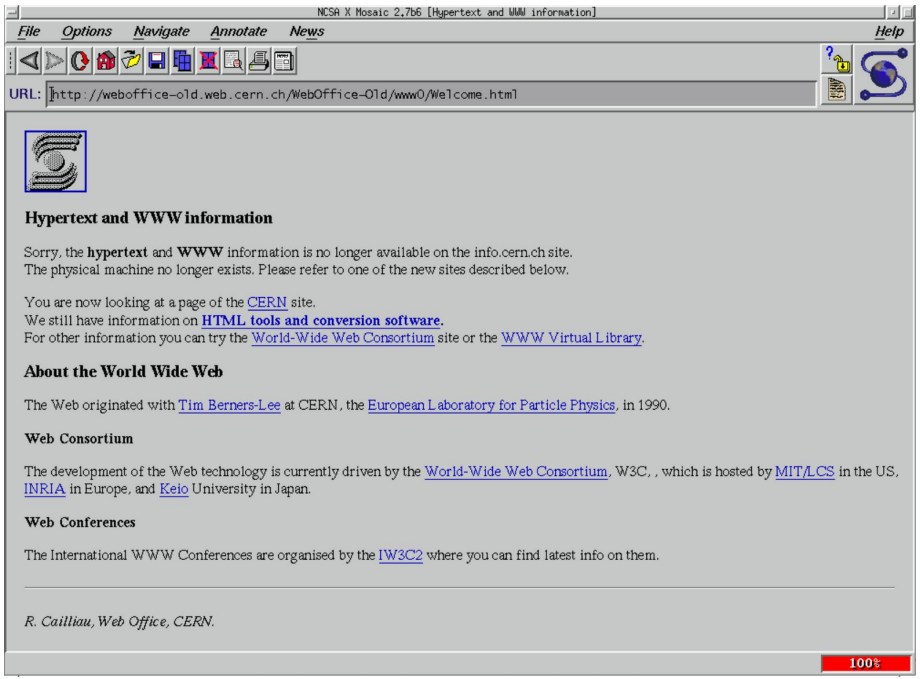
Cosa viene restituito dal seguente algoritmo se lo si esegue specificando il numero “4” come input?

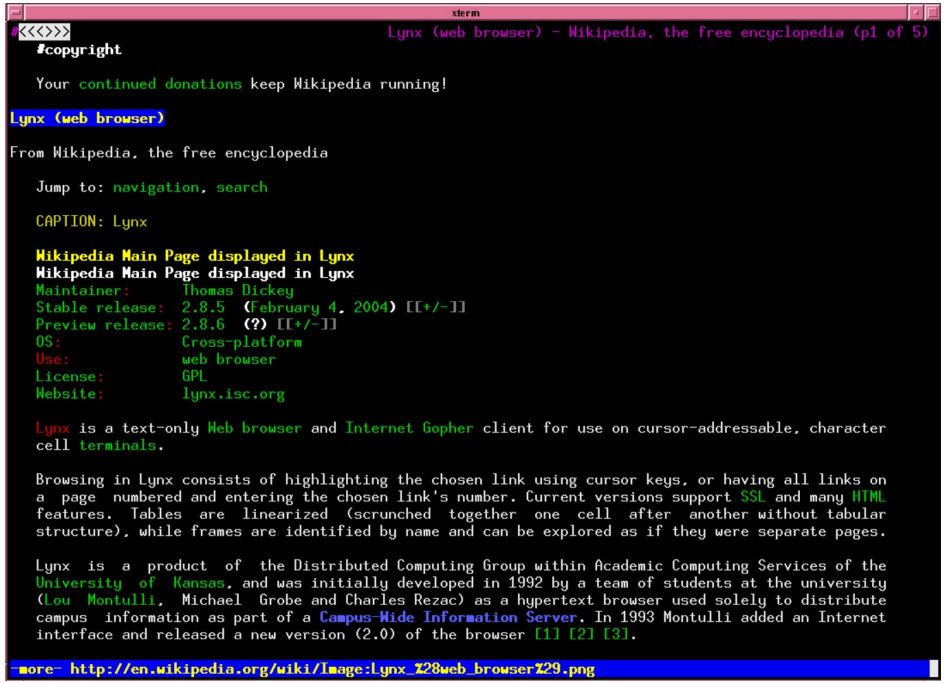
- ❏ 1
- ❏ 10
- ❏ 3
- ❏ 5



Browser testuali e grafici

Mosaic vs Lynx





Wrap-up

Le tecnologie informatiche nelle scienze umane

Informatica di base – a.a. 2020/2021

Silvio Peroni
0000-0003-0530-4305

Dipartimento di Filologia Classica e Italianistica, Università di Bologna, Bologna, Italia
silvio.peroni@unibo.it – @essepuntato – https://www.unibo.it/sitoweb/silvio.peroni/

DIPARTIMENTO DI FILOLOGIA CLASSICA E ITALIANISTICA