



UNIVERSITY OF TRENTO - Italy

Department of Information
and Communication Technology

Laboratorio 6

Pierluigi Roberti
Carmelo Ferrante

DISI – aa 2024/2025
Università degli Studi di Trento
pierluigi.roberti@unitn.it

Ripasso stringhe

char str[10];

ARRAY DA POSIZIONE 0 A 9

9 CARATTERI DISPONIBILI

LA VARIABILE «str» CONTIENE
L'INDIRIZZO DEL PRIMO ELEMENTO
DELL'ARRAY, QUELLO IN POSIZIONE 0

Array di caratteri

0	1	2	3	4	5	6	7	8	9
C	I	A	O						

Stringa

0	1	2	3	4	5	6	7	8	9
C	I	A	O	\0					

Ripasso stringhe

- Definizione di una stringa, cioè un array di caratteri che termina con '\0' (carattere di terminazione)

```
char stringa[20];  
char stringa[5]= "Ciao";  
char stringa[]= "Ciao";  
char stringa[]={ ' C ', ' i ', ' a ', ' o ', '\0'};
```

Definizioni ed
inizializzazioni
alternative

- Leggere una stringa da tastiera

```
fflush (stdin); scanf("%s", stringa);
```

oppure

```
fflush (stdin); gets(stringa);
```

- Stampare una stringa a video

```
printf("%s", stringa);
```

```
printf("Questa %s è una parola.\n", stringa);
```

Manca &!!!!

Inizializzazione array char

```
#define NCAR 20
```

```
char stringa[NCAR]; //19 caratteri utili
```

```
//primo modo (fine input invio o spazio)  
printf("stringa="); scanf("%s", stringa);
```

```
//secondo modo (fine input solo invio)  
printf("stringa="); gets(stringa);
```

```
//terzo modo - random
```

```
int i;
```

```
for(i=0; i<NCAR-1; i++) {
```

```
    stringa[i]=rand() % ('z' - 'a' + 1) + 'a';
```

```
}
```

```
stringa[NCAR-1]='\0'; //fine stringa
```

```
printf("stringa=%s \n", stringa);
```

Esercizio 1

Data una stringa, verificare se questa è parola palindroma oppure no. Lunghezza massima stringa **19** caratteri!

- Definire etichetta **NCAR** pari a **20** per dimensione array
- Definire stringa ed inizializzarla da tastiera
- Contare i reali caratteri contenuti nella stringa
- In base ai reali caratteri verificare se stringa è palindroma
- Stampare a video il risultato: esempio per **elle**, **alfa**
 - “La stringa **elle** è palindroma”
 - “La stringa **alfa** non è palindroma”

A	D	D	A
---	---	---	---

O	R	O
---	---	---

E	L	L	E
---	---	---	---

E	M	M	E
---	---	---	---

Modalità di confronto per verifica stringa palindroma

A	E	R	E	A
---	---	---	---	---

A	E	R	E	A
---	---	---	---	---

A	E	R	E	A
---	---	---	---	---

↑ ↑ ↑ ↑

↑ ↑ ↑ ↑

Soluzione esercizio 1 – ver 1

```
#include <stdio.h>
#include <stdlib.h>
#define NCAR 210

int main(void){
    char stringa[NCAR];
    int len=0;
    int i, j;
    int ispalindrome=1;
    printf("Inserire stringa: ");
    fflush(stdin);
    //scanf("%s",stringa);
    gets(stringa);
    //calcolo reale lunghezza stringa
    while(stringa[len]!='\0'){
        len++;
    }
    i = 0; //primo carattere
    j=len-1; //ultimo carattere

    while(i <= j && ispalindrome){
```

```
        if(stringa[i] != stringa[j])
        {
            ispalindrome = 0;
        }
        i++;
        j--;
    }

    if(ispalindrome){
        printf("La stringa %s e'
palindroma!\n", stringa);
    } else {
        printf("La stringa %s non e'
palindroma \n", stringa);
    }
    system("PAUSE");
    return 0;
}
```

Soluzione esercizio 1 – ver 2

```
char stringa[NCAR];
int len, i;
printf("stringa=");
fflush(stdin);
gets(stringa); //scanf("%s",stringa);
len=0;
while(stringa[len]!='\0') {
    len++;
}
printf("lunghezza=%d\n",len);
for(i=0; i<len/2; i++) {
    if(stringa[i]!=stringa[len-1-i]) {
        break;
    }
}
if(i==len) { printf("palindroma\n"); }
```

Soluzione esercizio 1 – ver 3

```
char stringa[NCAR];
int len, i, palindroma;
printf("stringa=");
fflush(stdin);
gets(stringa); //scanf("%s",stringa);
len=0;
while(stringa[len]!='\0') {
    len++;
}
printf("lunghezza=%d\n",len);
palindroma=1; //variabile sentinella
i=0;
while(i<len/2 && palindroma==1) {
    if(stringa[i]!=stringa[len-1-i]) {
        palindroma=0;
    }
    i++;
}
if(palindroma==1) { printf("palindroma\n"); }
```


Esercizio 2

Scrivere un programma che legge un carattere alla volta da tastiera e lo stampa immediatamente a video

- Eseguire il programma inserendo da tastiera dei caratteri generici
- Da sapere:
 - getchar() per leggere un singolo carattere alla volta (man getchar in un sistema GNU/Linux)
 - Per **leggere** dallo standard input un carattere alla volta
 - while ((c = getchar()) != EOF) { /* ...*/ }
 - A video si **stampa** usando la funzione
 - putchar(c)

NOTA: c va dichiarato come int poiché deve contenere anche il valore di EOF. Se dichiaro c char questo non potrebbe contenere EOF

- putchar converte l'int c in char e lo stampa

L'alfabeto del computer: il codice ASCII

- American Standard Code for Information Interchange
- Definito a partire dal 1963

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	 	Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	\$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	((72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051))	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[[123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174	|	
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135]]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177		DEL

Soluzione esercizio 2

```
#include <stdio.h>
```

```
int main (int argc, const char * argv[]) {  
    int c;  
    c = getchar();  
    while (c != EOF) {                // EOF => ctrl Z  
        putchar(c);  
        c = getchar();  
    }  
    return 0;  
}
```

Il carattere EOF viene ricevuto quando non c'è più input. Il nome ha più senso nel caso in cui l'input venga letto da un file reale, piuttosto che dall'input dell'utente (che è un caso speciale di un file).

ALTERNATIVA1:

```
while ((c = getchar()) != EOF) {  
    putchar(c);  
}
```

ALTERNATIVA2:

```
while (1) {  
    putchar(c);  
    c = getchar();  
}
```

Esercizio 3

Scrivere un programma che conta le occorrenze delle cifre, dei caratteri di spaziatura: spazio, carattere di tabulazione (\t) e new line (\n)

- Eseguire il programma inserendo da tastiera dei caratteri generici: si continua a leggere fino alla digitazione del carattere 'q'
- Esempio output

0: 1
1: 3
2: 0
3: 4
4: 1
5: 3
6: 2
7: 5
8: 5
9: 3

Definire un array `ndigit`
di dimensione 10 per il conteggio

Spazi: 12
Altri: 56

Definire 2 variabili `nwhite` e `nother`
per il conteggio

Esercizio 3

Conta Caratteri (numeri, caratteri, spazi, tab a capo):

- getchar() per leggere un singolo carattere alla volta
(man getchar in un sistema GNU/Linux)
- Per leggere dallo standard input un carattere alla volta
 - while ((c = getchar()) != 'q')
- Proprietà di rappresentazione alfanumerica delle cifre,
 - se (c >= '0' && c <= '9') il carattere c è una cifra
- Se il valore di c è carattere numerico
 - la cifra vale c – '0'

Soluzione esercizio 3

```
int c, i, nwhite, nother;
int ndigit[10];
nwhite = nother = 0;
for (i=0; i<10; i++){
    ndigit[i]=0;
}
while ((c=getchar()) != 'q'){
    if (c >= '0' && c <= '9') {
        ++ndigit[c - '0'];
    } else {
        if (c == ' ' || c == '\n' || c == '\t'){
            ++nwhite;
        } else {
            ++nother;
        }
    }
}
printf("cifre =");
for (i = 0; i < 10; ++i){ printf(" %d",ndigit[i]); }
printf(", spaziature = %d, altri = %d\n",nwhite,nother);
```

Ripasso Matrici (1/3)

Definizione di una matrice

```
#define NR 5
#define NC 4
int m[NR][NC];
```

	0	1	2	3
0				
1				
2				
3				
4				

Definizione ed inizializzazione di una matrice (valori costanti)

//inizializzazione completa della matrice

```
int m[NR][NC]={ {1,3,2,3}, {1,0,0,8}, {1,2,2,-1} };
int m[NR][NC]={1,3,2,3,1,0,0,8,1,2,2,-1};
```

Mancano dei valori

//inizializzazione parziale della matrice

```
int m[NR][NC]={ {1,3,2,3}, {1,0,0}, {1,2,2} };
```

//definizione parziale dimensioni della matrice

//inizializzazione completa della matrice

```
int m[][NC]={ {1,3,2,3}, {1,0,0,0}, {1,2,1,1} };
```

Ripasso Matrici (1/3)

```
int m[NR][NC]={ {1,3,2,3}, {1,0,0,8}, {1,2,2,-1}};
```

1	3	2	3
1	0	0	8
1	2	2	-1
0	0	0	0
0	0	0	0

```
int m[NR][NC]={1,3,2,3,1,0,0,8,1,2,2,-1};
```

1	3	2	3
1	0	0	8
1	2	2	-1
0	0	0	0
0	0	0	0

```
int m[NR][NC]={ {1,3,2,3}, {1,0,0}, {1,2,2}};
```

1	3	2	3
1	0	0	0
1	2	2	0
0	0	0	0
0	0	0	0

```
int m[][NC]={ {1,3,2,3}, {1,0,0,0}, {1,2,1,1}};
```

1	3	2	3
1	0	0	0
1	2	1	1
0	0	2	3
7541680	0	4199400	0

Ripasso Matrici (2/3)

Inizializzazione di una matrice (valori random)

```
    srand(time(0));  
    /*altrimenti ad ogni esecuzione stessa sequenza di  
    numeri pseudo casuali!  
    potrebbe essere necessario includere time.h*/  
    for(i=0 ; i<NR ; i++) {  
        for(j=0 ; j<NC ; j++) {  
            m[i][j] = rand() % (max - min + 1) + min;  
        }  
    }
```

Oppure (valori inseriti da utente)

```
    for(i=0 ; i<NR ; i++) {  
        for(j=0 ; j<NC ; j++) {  
            printf("valore: "); scanf("%d", &m[i][j]);  
        }  
    }
```

Ripasso Matrici (3/3)

Stampa del contenuto di una matrice

```
for(i=0 ; i<NR ; i++) {  
    for(j=0 ; j<NC ; j++) {  
        printf("%d ", m[i][j]); //"%3d"  
    }  
    printf("\n");  
}
```

Spazio dopo elemento



a capo dopo fine riga



Esercizio 4

Definire una matrice m di interi di dimensione NR (8) x NC (8)

```
#define NR 8
```

```
#define NC 8
```

- Inizializzare la matrice con numeri casuali compresi tra VMIN 0 e VMAX 100
- Stampare la matrice in modo opportuno
- Calcolare la media di tutti i valori della matrice
- Stampa a video la posizione delle celle della matrice m il cui valore è maggiore della media

Esercizio 5

Scrivere un programma che inizializza una matrice di 8 righe (NR) e 6 colonne (NC) contenente elementi di tipo intero.

- Inizializzare la matrice con numeri casuali compresi tra VMIN 18 e VMAX 30 (definire costanti).
- Stampare in modo opportuno la matrice
- Operazioni da implementare:

- Stampare i valori dell'array solo se sono >25

altrimenti stampare 'X'

- Cercare e stampare il valore minimo

- Cercare e stampare il valore massimo

- Calcolare la media

```
19 27 19 27
20 25 30 29
20 18 21 30
23 26 25 29
```

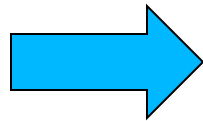
```
X 27 X 27
X X 30 29
X X X 30
X 26 X 29
```

```
media = 24.25
min = 18
max = 30
```

Esercizio 6

- Definire una matrice quadrata $NR \times NC$ (con $NR=NC=4$)
- Inserire da tastiera nella matrice dei valori compresi tra VMIN 0 e VMAX 9 .
 - Oppure inizializzarli in modo casuale
- Stampare a video la matrice in modo opportuno
- Modificare il contenuto della matrice in modo che:
 - ad ogni numero che è maggiore di 6 venga assegnato il valore 5.
 - ad ogni numero che è maggiore di 2 e minore di 7 venga assegnato il valore 0.
- Stampare a video la matrice

```
0 3 3 1
4 9 1 4
3 8 5 8
1 2 4 8
```



```
0 0 0 1
0 5 1 0
0 5 0 5
1 2 0 5
```

Esercizio 6 - bis

- Definire una matrice quadrata $NR \times NC$ (con $NR=NC=4$)
- Inserire valori casuali nella matrice, valori compresi tra VMIN 0 e VMAX 9
 - Oppure inizializzarli leggendoli da tastiera
- Stampare a video la matrice in modo opportuno
- Modificare il contenuto della matrice in modo che:
 - ad ogni **numero primo** sia assegnato il valore -1.
- Stampare a video la matrice risultante

0	3	3	1
4	9	1	4
3	8	5	8
1	2	4	8

-1	-1	-1	-1
4	9	-1	4
-1	8	-1	8
-1	-1	4	8

Esercizio 6 - ter

- Definire una matrice quadrata NRxNC (con NR=NC=4)
- Inserire in modo casuel nella matrice dei valori compresi tra 'A' e 'M'.

```
rand() % (max - min + 1) + min;
```

```
rand() % ('M' - 'A' + 1) + 'A';
```

- Oppure inizializzarli da tastiera
- Stampare a video la matrice in modo opportuno
- Modificare il contenuto della matrice in modo che:
 - ad ogni carattere compreso tra 'F' ed 'I' sia assegnato il carattere 'Z'.
- Stampare a video la matrice in modo opportuno

A	G	F	K
D	G	B	E
G	H	A	E
K	J	C	E

A	Z	Z	K
D	Z	B	E
Z	Z	A	E
K	J	C	E

Esercizio 6 extra

- Inserire da tastiera in un array bidimensionale denominato M(4x4) dei valori (compresi tra 0 e 9).
 - Oppure inizializzarli in modo casuale
- Verificare se esistono 2 numeri consecutivi uguali tra loro (esempio M[0][1] e M[0][2], in caso affermativo stampare il valore.
- Verificare se esistono 2 numeri consecutivi la cui somma è pari a 5 (esempio M[1][2]+M[1][3]), in caso affermativo stamparne il valore.

	0	1	2	3
0	2	2	3	1
1	4	9	1	4
2	3	8	5	8
3	1	2	4	8

Esercizio 7

Siano date due matrici A e B con le stesse dimensioni, calcolarne la somma:

- definire le costanti NR pari a 3 e NC pari a 2
- dichiarare ed inizializzare le matrici A e B
 - In modo statico (durante la dichiarazione)
 - In modo dinamico (con un ciclo assegnando valori casuali tra -5 e +5)
- dichiarare in modo opportuno una matrice S che conterrà la somma di A+B
- inizializzare la matrice S con contenuto pari a A+B
- stampare la matrice S

$$\begin{bmatrix} 1 & 3 & 2 \\ 1 & 0 & 0 \\ 1 & 2 & 2 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 5 \\ 7 & 5 & 0 \\ 2 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1+0 & 3+0 & 2+5 \\ 1+7 & 0+5 & 0+0 \\ 1+2 & 2+1 & 2+1 \end{bmatrix} = \begin{bmatrix} 1 & 3 & 7 \\ 8 & 5 & 0 \\ 3 & 3 & 3 \end{bmatrix}.$$

Esercizio 8

Siano date due matrici A e B tali che

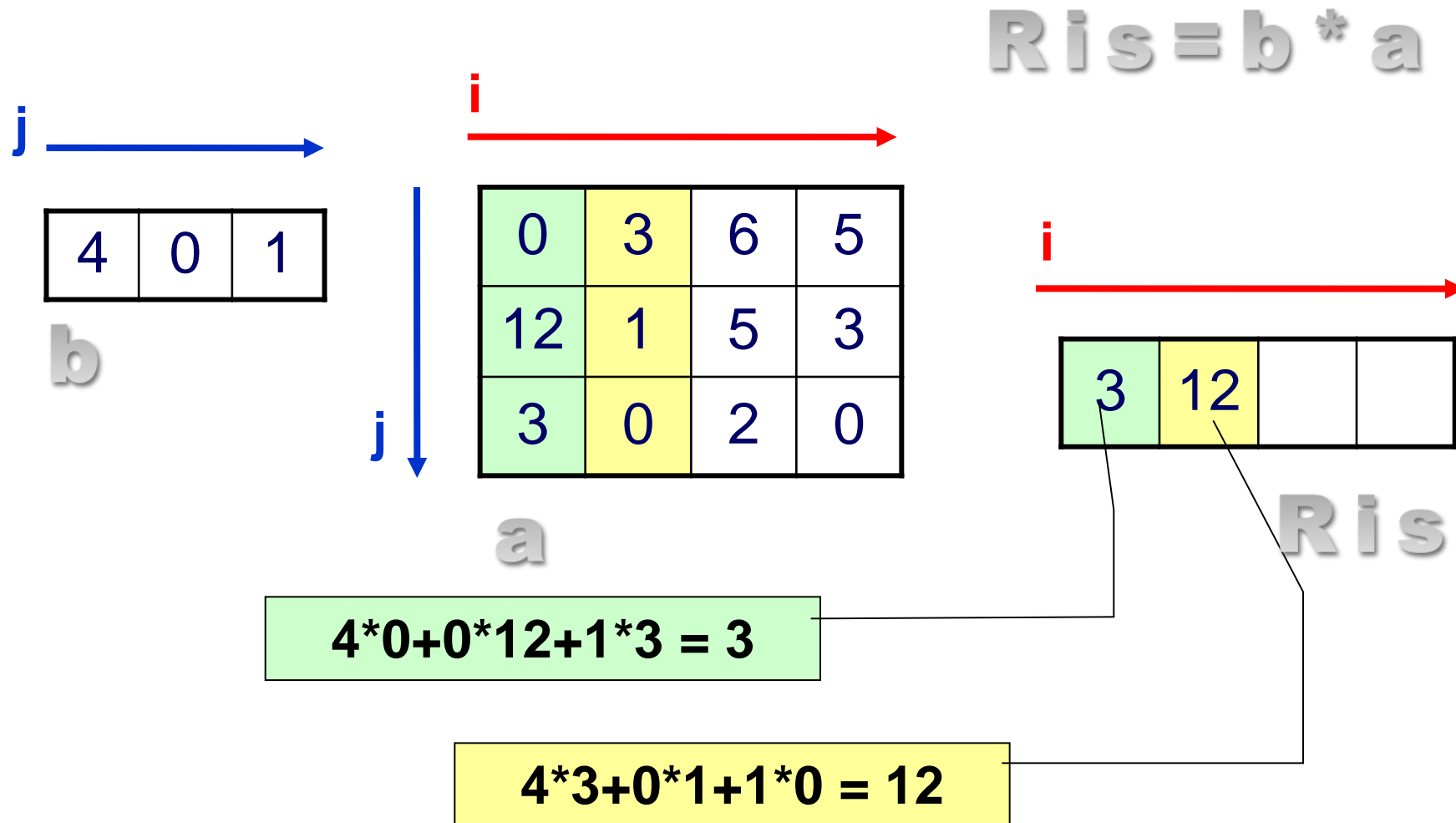
- A abbia **M** righe e **N** colonne
- B abbia **N** righe e **P** colonne
- Calcolare la matrice $C=A \times B$ di dimensione **M** x **P**, il cui elemento in posizione i,j è calcolato come:

$$C_{i,j} := A_{i,1}B_{1,j} + A_{i,2}B_{2,j} + \cdots + A_{i,n}B_{n,j}.$$

$$\begin{bmatrix} 1 & 0 & 2 \\ -1 & 3 & 1 \end{bmatrix} \times \begin{bmatrix} 3 & 1 \\ 2 & 1 \\ 1 & 0 \end{bmatrix} =$$

$$\begin{bmatrix} (1 \times 3 + 0 \times 2 + 2 \times 1) & (1 \times 1 + 0 \times 1 + 2 \times 0) \\ (-1 \times 3 + 3 \times 2 + 1 \times 1) & (-1 \times 1 + 3 \times 1 + 1 \times 0) \end{bmatrix} = \begin{bmatrix} 5 & 1 \\ 4 & 2 \end{bmatrix}$$

Esercizio 8



Esercizio 9

Scrivere un programma che popola una matrice $N \times N$ ($N=5$) con valori casuali tra 0 e VAL_MAX-1 .

Usare costanti ove possibile.

```
#define N 5
```

```
#define VAL_MAX 10
```

- Stampare la matrice
- Stampare l'indice della riga la cui somma è massima
 - Usare una variabile per tenere traccia dell'indice della riga la cui somma è massima ed una variabile per la relativa somma
- Stampare l'indice della colonna la cui somma è massima
 - Usare una variabile per tenere traccia dell'indice della colonna la cui somma è massima ed una variabile per la relativa somma