

Laboratorio di Programmazione
Corso di laurea triennale in Informatica Musicale
Università degli Studi di Milano, A.A. 2016-2017

Nicola Basilico, Andrea Lanzi

Appello del 31 Gennaio 2017

- L'esame ha una durata di 3 ore e 30 minuti.
- È possibile consultare il libro di testo, appunti e l'API Java.
- È possibile utilizzare qualsiasi ambiente di sviluppo installato sulla macchina.
- È proibito l'accesso ad Internet con qualsiasi mezzo.
- Ogni esercizio consegnato non deve generare errori di compilazione.
- Per ottenere la sufficienza è necessario consegnare almeno due esercizi corretti.

1 Quattro pari successivi a n

nome del file sorgente: PariSuccessivi.java

Si scriva un programma che legga un numero intero n e stampi a video i primi quattro numeri pari successivi a n

1.1 Esempio:

Dato in input n pari a:

11

il programma stampa:

12 14 16 18

Dato in input n pari a:

20

il programma stampa:

22 24 26 28

2 Massimo positivo e negativo

nome del file sorgente: MaxPositivoMaxNegativo.java

Si scriva un programma che legga in input una sequenza composta da 10 float. Si assuma che nella sequenza letta vi siano sempre almeno un numero positivo e almeno un numero negativo e che i numeri inseriti siano tutti diversi tra loro. Sia P il massimo (positivo) nella sequenza. Sia N il massimo negativo nella sequenza. Il programma stamperà:

1. quanti numeri compaiono tra N e P (o tra P e N) nella sequenza;
2. i numeri positivi tra N e P (o tra P e N) nella sequenza.

2.1 Esempio di esecuzione

Data in input questa sequenza:

```
23.5 -555.8 55.9 31.1 1200.5 42.7 -77.2 61.5 -12.2 124.7
```

il programma calcola $P=1200.5$ e $N=-12.2$ e stampa:

```
tra P e N ci sono 3 numeri, i positivi sono 42.7, 61.5.
```

Data in input questa sequenza:

```
-56.7 -10.2 11.7 34.2 -99.3 45.6 340.7 5.6 9.2 0.8
```

il programma calcola $P=340.7$ e $N=-10.2$ e stampa:

```
tra P e N ci sono 4 numeri, i positivi sono 11.7, 34.2, 45.6.
```

Data in input questa sequenza:

```
0.1 0.2 0.3 -0.1 1.5 -1.5 -0.2 -0.3 1 -1
```

il programma calcola $P=1.5$ e $N=-0.1$ e stampa:

```
tra P e N ci sono 0 numeri.
```

3 Controllo Password

nome del file sorgente: ControlloPassword.java

Il programma *ControlloPassword* legge una stringa inserita dall'utente ed effettua una sequenza di test per verificarne la robustezza. Come risultato stampa un rapporto che indica, per ogni test effettuato, se è stato superato o meno. Vengono effettuati quattro diversi test, ciascuno dei quali è definito come la verifica di una condizione sulla password inserita. I test verificano se:

1. la password è composta da almeno 5 caratteri (TEST 1);
2. la password contiene almeno una maiuscola e almeno una cifra decimale (0,1, ..., 9) (TEST 2);
3. tutti i caratteri in posizioni dispari sono consonanti (TEST 3);
4. i caratteri in posizioni pari formano una stringa palindroma (case sensitive) (TEST 4).

Si fornisca un'implementazione del programma che, una volta letta la password, stampi il rapporto e termini.

3.1 Esempio di esecuzione

L'utente inserisce questa password:

```
B6Gza4t6XHtHauGiB
```

Il programma stampa questo report:

```
TEST 1: ok
TEST 2: ok
TEST 3: fallito
TEST 4: ok
```

L'utente inserisce questa password:

```
1cGz
```

Il programma stampa questo report:

```
TEST 1: fallito
TEST 2: ok
TEST 3: ok
TEST 4: fallito
```

3.2 Info utili

Estratto della tabella ASCII.

CODICE	CARATTERE	DESCRIZIONE
32	(space)	(space)
33	!	(exclamation mark)
34	"	(Quotation mark)
35	#	(Number sign)
36	\$	(Dollar sign)
37	%	(Percent sign)
38	&	(Ampersand)
39	'	(Apostrophe)
40	((round brackets or parentheses)

CODICE	CARATTERE	DESCRIZIONE
41)	(round brackets or parentheses)
42	*	(Asterisk)
43	+	(Plus sign)
44	,	(Comma)
45	-	(Hyphen)
46	.	(Full stop , dot)
47	/	(Slash)
48	0	(number zero)
49	1	(number one)
50	2	(number two)
51	3	(number three)
52	4	(number four)
53	5	(number five)
54	6	(number six)
55	7	(number seven)
56	8	(number eight)
57	9	(number nine)
58	:	(Colon)
59	;	(Semicolon)
60	<	(Less-than sign)
61	=	(Equals sign)
62	>	(Greater-than sign)
63	?	(Question mark)
64	@	(At sign)
65	A	(Capital A)
66	B	(Capital B)
67	C	(Capital C)
68	D	(Capital D)
69	E	(Capital E)
70	F	(Capital F)
71	G	(Capital G)

CODICE	CARATTERE	DESCRIZIONE
72	H	(Capital H)
73	I	(Capital I)
74	J	(Capital J)
75	K	(Capital K)
76	L	(Capital L)
77	M	(Capital M)
78	N	(Capital N)
79	O	(Capital O)
80	P	(Capital P)
81	Q	(Capital Q)
82	R	(Capital R)
83	S	(Capital S)
84	T	(Capital T)
85	U	(Capital U)
86	V	(Capital V)
87	W	(Capital W)
88	X	(Capital X)
89	Y	(Capital Y)
90	Z	(Capital Z)
91	[(square brackets)
92	\	(Backslash)
93]	(square brackets)
94	^	(Caret or circumflex accent)
95	_	(underscore)
96	‘	(Grave accent)
97	a	(Lowercase a)
98	b	(Lowercase b)
99	c	(Lowercase c)
100	d	(Lowercase d)
101	e	(Lowercase e)
102	f	(Lowercase f)

CODICE	CARATTERE	DESCRIZIONE
103	g	(Lowercase g)
104	h	(Lowercase h)
105	i	(Lowercase i)
106	j	(Lowercase j)
107	k	(Lowercase k)
108	l	(Lowercase l)
109	m	(Lowercase m)
110	n	(Lowercase n)
111	o	(Lowercase o)
112	p	(Lowercase p)
113	q	(Lowercase q)
114	r	(Lowercase r)
115	s	(Lowercase s)
116	t	(Lowercase t)
117	u	(Lowercase u)
118	v	(Lowercase v)
119	w	(Lowercase w)
120	x	(Lowercase x)
121	y	(Lowercase y)
122	z	(Lowercase z)
123	{	(curly brackets or braces)
124		(vertical-bar)
125	}	(curly brackets or braces)
126	~	(Tilde)

4 k-Anagramma (con k=2)

nome del file sorgente: kAnagramma.java

Data una parola s e un intero k , un k -anagramma di s è una parola q dove ogni carattere di s compare k volte in q . Per esempio, se in una parola ci sono 3 “z” in un 2-anagramma di quella parola “z” dovrà comparire 6 volte. (Si noti come per

k=1 si ottenga la definizione classica di anagramma.) Si scriva un programma che, lette in input due stringhe, determini se la seconda è un 2-anagramma (case-sensitive) della prima.

4.1 Esempio di esecuzione

Date in input queste due stringhe:

```
na
anna
```

il programma stampa:

```
La seconda stringa è un 2-anagramma della prima.
```

Date in input queste due stringhe:

```
abcdefg
bbaaddeeffggcc
```

il programma stampa:

```
La seconda stringa è un 2-anagramma della prima.
```

Date in input queste due stringhe:

```
informatica
minorfatica
```

il programma stampa:

```
La seconda stringa non è un 2-anagramma della prima.
```

Date in input queste due stringhe:

```
ucm
uuccm
```

il programma stampa:

```
La seconda stringa non è un 2-anagramma della prima.
```