

# Esercizi per le vacanze di Pasqua- 4E

## Matrici di colori

Gli array bidimensionali possono essere utilizzati per vari scopi, per queste vacanze vorrei utilizzare queste strutture per approfondire i colori utilizzabili nella console di un qualsiasi programma in C++.

## Colori nella console C++

Il linguaggio C++ è in grado di gestire varie combinazioni di colori, noi, in particolare, ne utilizzeremo 256 utilizzando la seguente tabella:

256-color mode — foreground: ESC[38;5;#m background: ESC[48;5;#m																<a href="#">[hide]</a>
Standard colors								High-intensity colors								
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
216 colors																
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49
50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66
67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83
84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117
118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134
135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151
152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168
169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185
186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202
203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219
220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236
237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253
254	255															
Grayscale colors																

Immagine 1.1

Possiamo applicare questi colori sia

### Colore del testo

Il colore del testo si può cambiare utilizzando il codice **38** subito dopo il codice “\033[”

```
cout << "\033[38;5;" << COLORE << 'm' << TESTO_DA_VISUALIZZARE << "\033[m";
```

### Colore dello sfondo

Il colore del testo si può cambiare utilizzando il codice **48** subito dopo il codice “\033[”

```
cout << "\033[48;5;" << COLORE << 'm' << TESTO_DA_VISUALIZZARE << "\033[m";
```

Fatta questa piccola presentazione passare agli esercizi.

## Esercizi

1. Generare un array bidimensionale di `uint8_t` di dimensione 5 x 5 e riempirlo di valori casuali compresi nel range [0 - 255] tramite la funzione `rand()` presentata negli esercizi sugli array.

**Nota:** `uint8_t` è un tipo `int` di dimensione 8bit definito direttamente all'interno del linguaggio, quindi non è necessaria alcuna libreria per integrarlo.

In particolare, il seguente codice ci permette di accedere a tutte le celle della matrice e di assegnarne un valore:

```
uint8_t matrice[5][5];
for(int i = 0; i < 5; i++){
    for(int j = 0; j < 5; j++){
        matrice[i][j] = VALORE_MINORE_DI_255;
    }
}
```

1	2	3	22	2
9	123	12	12	214
8	34	0	52	23
15	88	62	88	67
234	123	3	53	7

- Visualizzare la matrice appena creata in output, vi consiglio di utilizzare il character literal '\t' fra un valore e l'altro per migliorarne la visualizzazione.

```
for(int i = 0; i < 5; i++){
    for(int j = 0; j < 5; j++){
        cout << '\t' << matrice[i][j];
    }
    cout << '\n';
}
```

- Stampare in output un'altra tabella seguendo le seguenti regole:
  - Il colore del testo deve essere nero.
  - Il colore dello sfondo deve corrispondere al valore presente nella matrice.

221	58	69	168	23	53	173	183	61	252
252	149	126	160	150	116	136	243	117	178
179	159	143	148	254	89	181	247	132	90
199	225	149	13	139	44	66	184	227	128
181	97	149	52	129	171	40	11	159	29
61	83	188	76	103	187	165	30	51	169
120	250	140	141	9	151	58	202	80	157
202	134	254	97	186	129	13	99	12	45
128	73	128	62	149	232	121	186	134	173
228	126	168	113	140	177	9	198	252	216

- Aggiungere all'inizio del valore stampato:
  - Uno '0' se il valore della cella è maggiore di 10 e minore di 100
  - Due '0' se il valore della cella è minore di 10

118	43	205	11	82	92	101	183	66	3
235	146	69	151	76	114	5	131	69	178
254	232	186	53	145	14	74	182	28	73
241	146	243	63	30	70	155	3	125	93
6	105	239	202	128	188	188	5	191	130
183	62	234	114	116	251	0	62	178	29
7	37	47	250	100	204	192	128	208	62
93	86	40	205	34	40	138	94	46	201
96	101	9	75	88	125	71	215	187	122
116	195	31	164	62	131	113	0	131	193

- Colorare il testo della matrice seguendo i colori utilizzati dall'[Immagine 1.1](#)

214	157	32	10	22	161	230	252	23	29
32	212	72	211	57	108	99	200	26	22
158	157	138	88	16	10	97	78	39	9
37	253	167	69	135	189	231	237	187	126
139	91	83	83	47	13	191	18	85	218
41	243	120	51	203	8	61	46	213	227
55	250	97	222	192	104	29	168	87	88
166	226	179	121	54	98	134	118	116	219
81	29	79	73	81	28	209	14	201	167
242	1	35	211	96	227	61	252	12	148

# Fonti

- [Stack Overflow 1](#)
- [Stack Overflow 2](#)
- [Wikipedia ANSI 8 bit](#)
- [Article](#)
- [Solution](#)

Consideriamo la matrice generata come un insieme di pixel ad [8bpp](#). Quindi, ogni valore salvato in questa tabella rappresenta un pixel con un valore che va da 0 a 255 secondo la seguente codifica:

Bit	7°	6°	5°	4°	3°	2°	1°	0°
Colore	R	R	R	G	G	G	B	B

Dove la riga Bit indica il bit preso in considerazione e la colonna Colore il colore corrispondente.

Nel caso del valore 212 abbiamo quindi che:

Decimale	Binario
212	11010100

Quindi secondo la tabella descritta precedentemente:

Bit	1	1	0	1	0	1	0	0
Colore	R	R	R	G	G	G	B	B

Avendo quindi che il rosso è presente nella quantità 6, il verde nella quantità 5 e il blu nella quantità 0.

Notare che:

- il rosso può assumere valori compresi nel range [0, 7]
- il verde può assumere valori compresi nel range [0, 7]
- il blu può assumere valori compresi nel range [0, 3]

A fianco si può vedere una tabella che riassume tutte le combinazioni possibili.