TD 02 – Tableaux et chaines de caractères

Exercice 1 Exécution mentale de code simple

- Pour chacun des extraits de code suivants, déterminez l'affichage dans la console.
  - Attention, il y a des pièges !

1.1

```
int a[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };
std::cout << std::size(a) << '\n';
for (int b : a)
  std::cout << b << ' ';
```

1.2

```
int a[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };
for (size_t i = 1; i < std::size(a); i += 2)
  std::cout << a[i] << ' ';
```

1.3

```
int a[10];
a[0] = 1;
for (size_t i = 1; i < std::size(a); i++)
  a[i] = a[i - 1] * 2;
for (int b : a)
  std::cout << b << ' ';
```

1.4

```
std::array a = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };
std::cout << a.size() << '\n';
for (int b : a)
{
  if (b == 5)
    break;
  std::cout << b << ' ';
}
```

1.5

```
std::array<unsigned int, 10> a = { 0, 1 };
std::cout << a.back() << '\n';
for (size_t i = 2; i < a.size(); i++)
  a[i] = a[i - 1] + a[i - 2];
std::cout << a.back();
```

1.6

```
std::vector a = { 1.0 };
std::cout << a.size() << ' ';
for (size_t i = 0; i < 4; i++)
  a.push_back(a.back() / 2.0);
std::cout << a.size() << '\n';
std::cout << a.back();
```

1.7

```
int a[2][3] = { {1,2,3}, {4,5,6} };
for (size_t i = 0; i < std::size(a); i++)
{
  for (size_t j = 0; j < std::size(a[i]); j++)
    std::cout << a[i][j] << ' ';
  std::cout << '\n';
}
```

1.8

```
int a[2][3] = { {1,2,3}, {4,5,6} };
for (size_t i = 0; i < 6; i++)
  std::cout << a[0][i] << ' ';
```

1.9

```
std::array<std::array<int, 3>, 2> a = { { {1,2,3}, {4,5,6} } };
for (auto b : a)
{
  for (auto c : b)
    std::cout << c << ' ';
  std::cout << '\n';
}
```

1.10

```
std::array<bool, 16> a;
uint16_t b = 43690;
for (int i = a.size() - 1; i >= 0; i--)
{
  a[i] = b % 2;
  b /= 2;
}
for (bool b : a)
  std::cout << b;
```

1.11

```
std::array<bool, 16> a = { 1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0 };
for (size_t i=0; i<4; ++i)
{
  char b = 8;
  char c = 0;
  for (size_t j = 0; j < 4; ++j)
  {
    if (a[i * 4 + j]) c += b;
    b /= 2;
  }
  if (c < 10) std::cout << static_cast<int>(c);
  else std::cout << static_cast<char>('A' + c - 10);
}
```

1.12

```
std::array<char, 4> a = { 10,10,10,10 };
uint32_t b = 0;
unsigned short c = 4096;
for (size_t i = 0; i < 4; ++i)
{
  b += c * a[i];
  c /= 16;
}
std::cout << b;
```

1.13

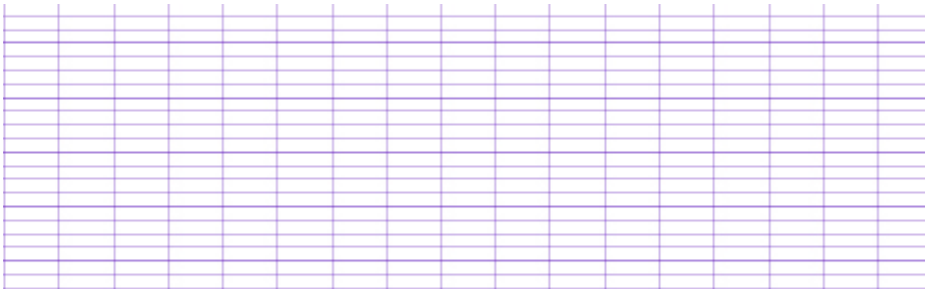
```
const size_t size = 5;
std::vector<std::vector<int>> t;
for (size_t i = 0; i < size; i++)
{
  std::vector<int> l;
  l.resize(i + 1);
  l[0] = 1;
  l[i] = 1;
  for (size_t j = 1; j < i; j++)
    l[j] = (t[i - 1][j - 1] + t[i - 1][j]);
  t.push_back(l);
}
for (auto l : t)
{
  for (auto p : l)
    std::cout << p << ' ';
  std::cout << '\n';
}
```

1.14

```
const size_t size = 10;
std::vector<std::vector<int>> s;
s.push_back({ 1 });
for (size_t i = 1; i < size; i++)
{
  std::vector<int> l;
  int cpt = 1;
  int lastn = s[i - 1][0];
  for (size_t j = 1; j < s[i-1].size(); j++)
  {
    if (s[i - 1][j] == lastn)
      cpt++;
    else
    {
      l.push_back(cpt);
      l.push_back(lastn);
      lastn = s[i - 1][j];
      cpt = 1;
    }
  }
  l.push_back(cpt);
  l.push_back(lastn);
  s.push_back(l);
}
for (auto l : s)
{
  for (auto p : l)
    std::cout << p << ' ';
  std::cout << '\n';
}
```

1.15

```
const size_t size = 5;
const size_t nbNiveaux = 8;
double distMax = std::sqrt((size / 2) * (size / 2) * 2);
std::vector<std::vector<size_t>> canvas(size);
for (size_t i = 0; i < size; ++i)
{
  canvas[i].resize(size);
  for (size_t j = 0; j < size; ++j)
  {
    double dist = std::sqrt(
      (size / 2 - i) * (size / 2 - i) +
      (size / 2 - j) * (size / 2 - j));
    canvas[i][j] = dist / distMax * nbNiveaux;
  }
}
for (auto l : canvas)
{
  for (auto p : l)
    std::cout << p;
  std::cout << '\n';
}
```



Exercice 2 Chaines de caractères

- Pour chacun des extraits de code suivants, déterminez l'affichage dans la console.
  - Attention, il y a des pièges !

2.1

```
char texte1[] = "Bonjour les gens !";
std::cout << "La taille de \"" << texte1 << "\" est " << std::size(texte1) << '\n';
char texte2[] = "Bonjour les étudiant·e·s !";
std::cout << "La taille de \"" << texte2 << "\" est " << std::size(texte2) << '\n';
```

2.2

```
char texte[] = "Bonjou";
texte[6] = 'r';
std::cout << texte << '\n';
```

2.3

```
char texte[] = "Bonjour";
size_t i = 0;
while (texte[i])
{
  if (texte[i] >= 'a' && texte[i] <= 'z')
    texte[i] += 'A' - 'a';
  ++i;
}
std::cout << texte;
```

2.4

```
char texte_classique[] = "Bonjour";
std::string texte_string = "Bonjour";
std::cout << "La taille de \"" << texte_classique
          << "\" est " << std::size(texte_classique) << '\n';
std::cout << "La taille de \"" << texte_string << "\" est "
          << texte_string.size() << '\n';
```

2.5

```
std::array<std::string, 12> mois = { "janvier", "février",
         "mars", "avril", "mai", "juin", "juillet", "août",
         "septembre", "octobre", "novembre", "décembre" };
unsigned int j, m, a;
std::cin >> j >> m >> a;
std::string jour = std::to_string(j);
if (j == 1)
  jour += "er";
std::string date = jour + ' ' + mois[m - 1] + ' ' +
std::to_string(a);
std::cout << date <<'\n':
```

- On supposera que l'utilisateur a saisi « **1 10 2022** »

2.6

```
std::string texte1 = "Alphabet";
std::string texte2 = "zoo";
if (texte1 < texte2)
  std::cout << texte1 << " est avant " << texte2 << '\n';
else
  std::cout << texte1 << " est après " << texte2 << '\n';
texte1 = "alphabet";
texte2 = "Zoo";
if (texte1 < texte2)
  std::cout << texte1 << " est avant " << texte2 << '\n';
else
  std::cout << texte1 << " est après " << texte2 << '\n';
```

2.7

```
const char palette[] = "Wwli:,. ";
std::array bitmap = { 7,6,5,6,7,6,4,2,4,6,5,2,0,2,5,6,4,2,4,6,7,6,5,6,7 };
size_t size = std::sqrt(bitmap.size());
for (size_t i = 0; i < size; ++i)
{
  for (size_t j = 0; j < size; ++j)
    std::cout << palette[bitmap[i * size + j]];
  std::cout << '\n';
}
```

2.8

```
int n;
float r, i, R, I, b;
for (i = -1; i < 1; i += .2, std::cout<<'\n')
  for (r = -2; I = i, (R = r) < 1; r += .1, std::cout<<static_cast<char>(n + 31))
    for (n = 0; b = I * I, 26 > n++ && R * R + b < 4; I = 2 * R * I + i, R = R * R - b + r);
```