

1. Code d'une fonction `identite` qui renvoie la matrice identité (ici type `float` mais d'autres types, par exemple `int`, étaient possibles) :

```
float* identite(int n)
{
    float* M = malloc(n*n*sizeof(float));
    for(int i=0;i<n;i++)
        for(int j=0;j<n;j++)
        {
            if (i==j)
                M[i*n+j]=1;
            else
                M[i*n+j]=0;
        }
    return M;
}
```

2. Code d'une fonction `symetrique` qui renvoie 1 si la matrice M passée en paramètre est symétrique (*i.e.*, carrée avec $M_{ij} = M_{ji}$), 0 sinon (on pouvait aussi renvoyer un booléen ou écrire un message) :

```
int symetrique(float* M,int p, int q)
{
    if (p!=q) return 0;
    for(int i=0;i<p;i++)
        for(int j=0;j<q;j++)
            if (M[i*p+j]!=M[j*p+i]) return 0;
    return 1;
}
```

Remarque : la boucle sur j pourrait s'arrêter à $j < i$.

3. La fonction `identite` ci-dessus effectue n^2 affectations : sa complexité asymptotique est $O(n^2)$. La fonction `symetrique` ci-dessus effectue au plus $1+p \times q$ tests : sa complexité asymptotique est $O(n^2)$ avec $n = p = q$.