

```
// Solutions de l'exercice 6 de l'examen 2017
```

```
#include <iostream>
```

```
using namespace std;
```

```
void affiche(int , int[]);
```

```
int create_valide(int , int[], int[]);
```

```
int remplacer(int, int[], int , int );
```

```
int main()
```

```
{
```

```
    const int NDIM=21;
```

```
    int notes[NDIM]={13,-1,0,19,10,0,17,9,-1,2,6,9,0,4,12,14,18,3, 5,6,-1} ;
```

```
    int valides[NDIM], nomb_notes_valides,nomb_remp;
```

```
    nomb_notes_valides = create_valide(NDIM,notes,valides);
```

```
    cout << "Le tableau de base a " << NDIM << " valeurs, mais il n y a que "
```

```
    << nomb_notes_valides << " notes valides " << endl;
```

```
    cout << "Voici le tableau de notes valides " << endl;
```

```
    affiche(nomb_notes_valides,valides);
```

```
    nomb_remp = remplacer(NDIM, notes, -1, 0);
```

```
    cout <<endl<< "En remplaçant les " << nomb_remp << " valeur de -1 par 0 on obtient un tableau de  
notes comme suit " << endl;
```

```
    affiche(NDIM,notes);
```

```
    return 0;
```

```
}
```

```
void affiche(int ndim, int tab[])
```

```
{
```

```
    for(int i=0;i<ndim;i++)
```

```
    {
```

```
        cout << "L'element " << i << " du tableau = " << tab[i] << endl;
```

```
    }
```

```
}
```

```
int create_valide(int ndim, int tab[], int tabvalide[])
```

```
{
```

```
    int ivalid = 0;
```

```
    for(int n=0; n<ndim ; n++)
```

```
    {
```

```
        if(tab[n] >= 0)
```

```
        {
```

```
            tabvalide[ivalid] = tab[n]; ivalid++;
```

```
        }
```

```
    }
```

```
    return ivalid;
```

```
}
```

```
int remplacer(int ndim, int tab[], int val_old, int val_new) /* fonction remplace les valeurs val_old par  
des valeurs val_new */
```

```
{
```

```
    int nrepl = 0;
```

```
    for(int n=0; n<ndim ; n++)
```

```
    {
```

```
        if(tab[n]==val_old) tab[n] = val_new;
```

```
        nrepl++;
```

```
}  
return nrempl;  
}
```