

3. Qu'affiche le code suivant :

```
#include<stdio.h>
int
main
(void)
{
    int s=0;
    while(s++<10)
    {
        if(s<4 && s<9)
            continue;
        printf(" %d ",s);
    }
    return s ;
}
```

break → fait sortir le bloc
continue → mais affiche qd
c'est vérifié

4, 5, 6, 7, 8, 9, 10

4. Donnez la définition de la fonction de prototype

```
int cherche_et_incremente (int tab[], int size) ;
```

qui recherche et incrémente la valeur maximale d'un tableau tab de taille size; de plus, cette fonction doit retourner la valeur minimale du tableau.

5. Soient les définitions suivantes :

```
char *t1[4] = { "Il est beau", "le lavabo", "Il est laid", "le bidet" } ;
int t2[9] = { 0, 2, 4, 6, 7, 1, 456, 24, 4 } ;
int *p = &t2[4] ;
```

Donnez le type et la valeur retournée par l'évaluation des expressions suivantes :

t1[2] *Il est laid => char**
t1[2][3]
*t1
t2[2]
*t2
*p
(*p)+2
*(p+2)
t1[t2[2]]

2 Matrices pleines de dimensions variables

La taille d'une matrice pleine de dimensions variables n'est pas connue à la compilation mais seulement lors de sa création. Les coefficients de cette matrice sont des entiers machines signés.

On se donne la description des types suivante :

- le type `matrix_t` est un pointeur sur des objets définis suivant le modèle d'identificateur `matrix_m`;

un tel objet est :

- soit un objet d'identificateur `zero` étant vrai si la matrice est nulle et faux dans le cas contraire,

- soit un objet suivant le modèle d'identificateur `truematrix_m`.

- un objet de la famille d'identificateur `truematrix_m` est composé :

- d'un entier signé d'identificateur `nblig` codant le nombre de lignes;

- d'un entier signé d'identificateur `nbcol` codant le nombre de colonnes;

- d'un pointeur sur des entiers signés d'identificateur `body`.

matrix_m

truematrix_m

()