Ex1

An industrial carpentry manages a stock of wooden panels. Each panel has a width, length and thickness in millimeters, as well as the type of wood which can be pine (code 0), oak (code 1) or beech (code 2).
Write an algorithm that
Define a Panel structure containing all the information relating to a wooden panel
Entering and displaying a wooden panel.
Calculate the volume in cubic meters of a panel

```
typedef struct bois
{
float largeur, longueur, epaisseur;
char essence;
} Panneaux;
```

b)

```
Panneaux Saisie()
{
Panneaux p;
printf("Entrez la largeur, la longueur et l'épaisseur : ");
scanf("%f %f %f", &p.largeur, &p.longueur, &p.epaisseur);
printf("Entrez l'essence de bois : ");
scanf("%c", &p.essence);
return p;
}
void Affichage(Panneaux p)
{
printf("Panneau en ");
switch (p.essence)
{
case '0':
printf("pin\n");
break;
case '1':
printf("chêne\n");
break;
case '2':
printf("hêtre\n");
break;
default:
printf("inconnue\n");
}
printf("largeur = %f ; longueur = %f ; epaisseur = %f\n",
p.largeur, p.longueur, p.epaisseur);
}
```

c)

```
float Volume(Panneaux p)
{
return (p.largeur * p.longueur * p.epaisseur) / 1e9;
}
```

Ex#2

Write a program that allows you to fill a table of dimension N with patient information which is: the patient's name, weight (integer, in kg), height (integer, in cm) (N entered by the user); the program then displays the number of obese patients.
Fat mass is assessed from the calculation of the body mass index or BMI which takes into account weight and height.

$$BMI = \frac{Weight\ in\ kg}{(heigt\ in\ meters)^2}$$

BMI=(weight in kg)/((height in meters)²); $BMI \geq 30 \rightarrow Obesity$

```
#include<stdio.h>
```

```
typedef  struct {
        char nom[30];
                int poids;
                int taille;
        }patient;
main()
{       patient listp[100];
        int i , n , nbob;
        float IMC ;
        printf (" donner le nombre de patients :");
        scanf("%d",&n);
        printf ("entrer les infos de chaque patient : \n");
        for(i=0;i<n;i++)
        {       printf ("patient %d \n", i+1);
                printf (" nom : ");
                scanf("%s",listp[i].nom);
                printf (" poids : ");
                scanf("%d",&listp[i].poids);
                printf (" taille : ");
                scanf("%d",&listp[i].taille);
        }
        nbob=0;
        for(i=0;i<n;i++)
                IMC=listp[i].poids/((listp[i].taille/100.0) * (listp[i].taille/100.0));
                        if (IMC >30)
                        nbob++;
        printf(" nombre de patients obèses  :%d",nbob);
}
```

Write a program which generates an array V of n integers then arranges the non-zero elements of the array in an array of structure T; a structure contains 3 fields defined according to the following principle:
- ValAbs: is the absolute value of a non-zero element of table V.
- Pos: its position (index).
- Sign: its sign; either "+" for a positive element or "-" for a negative element.

Finally the program iterates through T and displays these positive elements and their sum.
Example: if the user reads the following table V:

0	1	2	3	4	5	6	7
3	0	-8	-4	0	0	0	9

the program constructs the following table T:

0			1			2			3		
3	0	'+'	8	2	'-'	4	3	'-'	9	7	'+'
ValAbs	*pos*	*signe*	*ValAbs*	*pos*	*signe*	*ValAbs*	*pos*	*signe*	*ValAbs*	*pos*	*signe*

The program displays:
The positive elements of T are: 3 9 their sum: 12

```
#include <stdio.h>
#include <math.h>
typedef struct {
        int ValAbs;
        int pos;
        char signe;
```

```
        }Element ;
main()
{
 /* Déclarations */
  int n, i ,j,som;     /* dimension        */
 int v[50];
 Element T[50];
  /* Saisie des données */
 printf("Dimension des tableaux  : ");
 scanf("%d", &n );

 printf("** entrer les éléments de v :  **\n");
 for (i=0; i<n; i++)
    {     printf("Element %d : ", i);
     scanf("%d", &v[i]);
    }
    j=0;
for (i=0; i<n; i++)
    if (v[i]!=0)
    {          T[j].ValAbs=abs(v[i]);
               T[j].pos=i;
               if (v[i]>0)
                   T[j].signe='+';
               else
                   T[j].signe='-';
               j++;
      }
 printf (" les éléments positifs du tableau : \n ");
 som=0 ;
 for (i=0;i<j;i++)
   if (T[i].signe=='+')
        { printf ("%d \t",T[i].ValAbs)   ;
          som=som+ T[j].ValAbs;
         }
 rintf ("leur somme : %d ",som)   ;
}
```