

---

## COMPTE RENDU TP 3 RN40

BARTHELME ALEXANDRE

---

Mon code complet est disponible sur mon GitHub à l'adresse :  
<https://github.com/Iron68A/RN40/tree/main/TP/TP3>

Ou par capture d'écran

### I) Le calcul du 2<sup>nd</sup> degré

Fonction qui prend en arguments les 3 coefficients, et un pointeur vers le nombre de solutions et les solutions ( n ,x1,x2 ).

Elle calcul le delta ( $b^2-4ac$ ) puis les racines. Si delta est négatif, la fonction s'arrête. Sinon elle renvoi les résultats des racines

### II) La liste de produit

On a définit dans le header ce qu'est un produit ( code, quantité , prix) puis on en a fait une liste de produits. Toutes les opérations sur cette liste sont en captures d'écrans.

### III) Captures d'écran :

Calcul du 2<sup>nd</sup> degré :

```
void seconddeg(float a, float b, float c,int *n, float *x1, float*x2){
    printf("la fonction est %f*x^2+%f*x+%f \n" ,a,b,c);
    float delta=(b*b)-(4*a*c);
    if (delta<0) *n=0;
    else if (delta==0) {
        *n=1;
        *x1=(-1*b)/(2*a);
    }
    else {
        *n=2;
        *x1=((-1*b)-sqrtf(delta))/(2*a);
        *x2=((-1*b)+sqrtf(delta))/(2*a);
    }
}
```

Liste de produits :

```
List ajout_tete(List l, int c, int q, float p){
    List l2 = (prod * )malloc(sizeof(prod));
    l2->codeP = c;
    l2->quantP = q;
    l2->prixP = p;
    l2->next = l;
    return l2;
}
```

```
//ajout en fin de liste
List ajout_fin(List l, int c, int q, float p){
    if (l == NULL){
        ajout_tete(l, c, q, p);
    }
    else{
        List l2 = (prod * )malloc(sizeof(prod));
        l2->codeP = c;
        l2->quantP = q;
        l2->prixP = p;
        l2->next = NULL;
        List l3 = l;
        while (l3->next != NULL){
            l3 = l3->next;
        }
        l3->next = l2;
        return l;
    }
}
```

```

//suppression premier element de liste
List supp_tete(List l){
    if (l == NULL) return l;
    else {
        List l2 = l;
        l = l->next;
        free(l2);
        return l;
    }
}

//suppression dernier de liste
List supp_fin(List l){
    if (l == NULL) return l;
    else {
        List l2 = l;
        List l3 = l;
        while (l2->next != NULL){
            l3 = l2;
            l2 = l2->next;
            l3->next = NULL;
        }
        free(l2);
        return l;
    }
}

```

```

void afficher_list(List l){
    printf(" Code | Quantite | Prix\n");
    printf("-----\n");
    while(l != NULL){
        printf(" %d | %d | %f\n", l->codeP, l->quantP, l->prixP);
        l = l->next;
    }
    printf("-----\n");
}

//recherche d'un produit via son code
List recherche(List l, int c){
    while (l != NULL){
        if (l->codeP == c){
            return l;
            printf("produit trouve\n");
            printf("codeP = %d, quantP = %d, prixP = %f\n", l->codeP, l->quantP, l->prixP);
        }
        l = l->next;
    }
    return NULL;
}

```