

Exercices supplémentaires sur les structures répétitives

Exercice 51 :

Ecrire un programme qui demande un nombre de départ à l'utilisateur, et qui affiche les 10 nombres suivants. Par exemple si l'utilisateur entre le nombre 19, l'algorithme affichera les nombres de 20 à 29.

Réponse :

```
#include <stdio.h>
#include <stdlib.h>
int main(){
    int n,i,a;
    printf("Donner un nombre entier: ");
    scanf("%d",&n);
    printf("Les 10 nombres suivantes sont: ");
    for(i=1;i<=10;i++)
        printf("%d ",n+i);
    return 0;
}
```

Exercice 52 :

Ecrire un programme permettant de calculer le produit de deux nombres entiers positifs donnés par l'utilisateur sans utiliser l'opérateur de multiplication.

Réponse :

```
#include <stdio.h>
#include <stdlib.h>
int main(){
    int a,b,p,i;
    printf("Donner deux nombres entiers positifs: ");
    scanf("%d%d",&a,&b);
    if(a<0 || b<0)printf("Erreur de saisie");
    else{
        if(a==0 || b==0)p=0;
        else{
            p=0;
            for(i=1;i<=b;i++)
                p=p+a;
        }
        printf("Le produit est: %d",p);
    }
    return 0;}
}
```

Exercice 53 :

Ecrire un programme qui demande à l'utilisateur un nombre N , puis calcule la somme des N premiers carrés en affichant tous les résultats intermédiaires.

Exemple: avec $N = 4$, Somme = $1^2+2^2+3^2+4^2= 30$.

Réponse :

```
#include <stdio.h>
#include <stdlib.h>
int main(){
    int n,s,i;
    printf("Donner un nombre entier positif: ");
    scanf("%d",&n);
    if(n<=0)printf("Erreur de saisi");
    else{
        s=0;
        for(i=1;i<=n;i++){
            s=s+i*i;
            printf("s=%d\n",s);
        }
    }
    return 0;
}
```

Exercice 54 :

Voici l'énoncé d'un jeu simple. On initialise d'abord un nombre entre 0 et 20 aléatoirement. Ensuite, l'utilisateur est interrogé pour prédire ce nombre. En cas de réponse erroné, l'algorithme doit aider l'utilisateur en lui affichant un message : «Le nombre caché est plus petit » ou bien «Le nombre caché est plus grand ». Lorsque l'utilisateur arrive à prédire le nombre caché un message de félicitation est lui affiché. Ecrire le programme résolvant ce problème.

Remarque : pour générer un nombre aléatoirement en langage C, on utilise la fonction prédéfinie `rand()` de la bibliothèque `<windows.h>`. Cette utilisation doit être précédée par l'initialisation de `rand()` à l'aide de l'instruction `srand(time(NULL))`.

Réponse :

```
#include <stdio.h>
#include <stdlib.h>
#include <windows.h>
int main(){
    int nbCache,v;
    srand(time(NULL));
    nbCache=rand()%20;
    do{
        printf("Veuillez predire le nombre cache: ");
        scanf("%d",&v);
        if(nbCache>v)printf("Le nombre cache est plus grand\n");
        else if(nbCache<v)printf("Le nombre cache est plus petit\n");
    }while(nbCache!=v);
    printf("Bravo..vous avez trouve le nombre cache");
    return 0;
}
```

Exercice 55 :

Modifier le programme de l'exercice précédent pour qu'il n'autorise que 5 tentatives. Si l'utilisateur n'arrive pas à prédire le nombre caché dans 5 tentatives, le message « Echec, jeu terminé. » lui est affiché conjointement avec le nombre caché et avec un avertissement sonore.

Réponse :

```
#include <stdio.h>
#include <stdlib.h>
#include <windows.h>
int main(){
    int nbCache,v,nbTent;
    srand(time(NULL));
    nbCache=rand()%20;
    printf("Veuillez predire le nombre cache\n");
    printf("Vous avez le droit de 5 tentatives\n");
    nbTent=1;
    do{
        printf("Tentative %d: ",nbTent);
        scanf("%d",&v);
        nbTent=nbTent+1;
        if(nbCache>v)printf("Le nombre cache est plus grand\n");
        else if(nbCache<v)printf("Le nombre cache est plus petit\n");
    }while(nbCache!=v && nbTent<=5);
    if(nbCache==v)printf("Bravo..vous avez trouve le nombre cache");
    else printf("Echec,jeu termine.Le nombre cache est: %d\a",nbCache);
    return 0;
}
```

Exercice 56 :

Ecrire un programme qui demande à l'utilisateur un nombre entier positif non nul et détermine si celui-ci est un nombre premier ou pas.

Rappelons qu'un nombre premier est un nombre dont ses seuls diviseurs sont 1 et lui-même.

Exemple : 2, 3, 5, 7, 11, 13, 17, 19 ... sont des nombres premiers.

Réponse :

```
#include <stdio.h>
#include <stdlib.h>
int main(){
    int n,i;
    printf("Donner un nombre entier positif: ");
    scanf("%d",&n);
    if(n<=0)printf("Erreur de saisi");
    else{
        i=2;
        while(i<=n-1 && n%i!=0)i=i+1;
        if(i>=n)printf("%d est un nombre premier",n);
        else printf("%d n'est pas un nombre premier",n);
    }
    return 0;
}
```

Exercice 57 :

Ecrire un programme permettant à l'utilisateur de saisir les notes d'une classe, une par une, et qui s'arrête lorsqu'on rentre une note négative ou supérieure à 20 (ce qui signifie qu'il n'y a plus de note). Le programme, une fois la saisie est terminée, renvoie la moyenne des notes de la classe, le nombre des notes supérieures ou égales à dix, la note minimale et la note maximale dans la classe.

Réponse :

```
#include <stdio.h>
#include <stdlib.h>
int main(){
    float note,moy,min,max,somme;int nbNotes,nbSup10;
    printf("Donner une note: ");
    scanf("%f",&note);
    if(note <0 || note >20)printf("Pas de notes");
    else{
        min=note;max=note;
        nbNotes=0; nbSup10=0;
        somme=0;
        while(note>=0 && note<=20){
            nbNotes=nbNotes+1;
            if(note>=10)nbSup10=nbSup10+1;
            if(note>max)max=note;
            else if(note<min)min=note;
            somme=somme+note;
            printf("Donner une autre note: ");
            scanf("%f",&note);
        }
        moy=somme/nbNotes;
        printf("La moyenne des notes est: %.2f\n",moy);
        printf("Le nombre des notes >=10 est: %d\n",nbSup10);
        printf("La note minimale est: %.2f\n",min);
        printf("La note maximale est: %.2f\n",max);
    }
    return 0;
}
```

Exercice 58 :

Ecrire un programme permettant de calculer la puissance deux nombres entiers positifs sans avoir utiliser la fonction prédéfinie `pow()` de la bibliothèque `<math.h>`.

Réponse :

```
#include <stdio.h>
#include <stdlib.h>
int main(){
    int a,b,p,i;
    printf("Donner 2 nombres entiers positifs: ");
    scanf("%d%d",&a,&b);
    if(a<0 || b<0)printf("erreur de saisi");
    else{
        if(b==0)p=1;
        else {
            p=1;
            for(i=1;i<=b;i++)
                p=p*a;
        }
    }
}
```

```

    }
    printf("La puissance est: %d",p);
}
return 0;
}

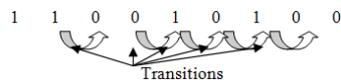
```

Exercice 59 :

Il vous est demandé dans cet exercice d'établir un programme permettant de saisir les états (sous forme de 0 et de 1 ; par exemple 0 : repos, 1 : en mouvement) d'un objet physique dans 10 instants différents, et de calculer le nombre de transitions. Une transition est le changement d'état entre deux instants successifs ; en d'autres termes le passage de 0 à 1 ou bien de 1 à 0.

Le programme doit assurer que les valeurs entrées sont correctes (uniquement des 0 et des 1).

Exemple : le nombre de transitions pour les états suivantes est : 5



Réponse :

```

#include <stdio.h>
int main(){
    int etat,nbTran,i,etatPrec;
    printf("Donner l'etat 1 (0/1): ");
    scanf("%d",&etat);
    while(etat!=0 && etat!=1){
        printf("Erreur, redonner l'etat 1 (0/1): ");
        scanf("%d",&etat);
    }
    etatPrec=etat;nbTran=0;
    for(i=2;i<=10;i++){
        printf("Donner l'etat %d (0/1): ",i);
        scanf("%d",&etat);
        while(etat!=0 && etat!=1){
            printf("Erreur, redonner l'etat %d (0/1): ",i);
            scanf("%d",&etat);
        }
        if(etat!=etatPrec)nbTran=nbTran+1;
        etatPrec=etat;
    }
    printf("Le nombre de transitions est: %d",nbTran);
    return 0;
}

```

Exercice 60 :

Ecrire un programme qui demande à l'utilisateur un nombre entier positif non nul et :

- Affiche les diviseurs de ce nombre
- Le nombre de ces diviseurs
- La somme des diviseurs de ce nombre

Réponse :

```

#include <stdio.h>
#include <stdlib.h>
int main(){
    int n,i,sDiv,nbDiv;
    printf("Donner un nombre entier positif: ");
    scanf("%d",&n);
    if(n<=0)printf("erreur de saisi");

```

```

else{
    sDiv=0;nbDiv=0;
    printf("Les diviseurs de %d sont: \n",n);
    for(i=1;i<=n;i++){
        if(n%i==0){
            printf("%d ",i);
            nbDiv=nbDiv+1;
            sDiv=sDiv+i;
        }
    }
    printf("\nLe nombre des diviseurs est: %d\n",nbDiv);
    printf("La somme des diviseurs est: %d",sDiv);
}
return 0;
}

```

Exercice 61 :

Ecrire un programme qui lit n nombres entiers (n est donné par l'utilisateur) et détermine combien de fois la première valeur a été saisie (sans compter la première saisie).

Réponse :

```

#include <stdio.h>
#include <stdlib.h>
int main(){
    int n,v,prem,nb,i;
    printf("Donner le nombre de valeurs a saisir: ");
    scanf("%d",&n);
    if(n<=0)printf("Erreur de saisi");
    else{
        printf("Donner la premiere valeur: ");
        scanf("%d",&prem);
        nb=0;
        for(i=2;i<=n;i++){
            printf("Donner la %deme valeur: ",i);
            scanf("%d",&v);
            if(v==prem)nb=nb+1;
        }
        printf("%d a ete saisi %d fois",prem,nb);
    }
    return 0;
}

```

Exercice 62 :

Ecrire un programme permettant de lire un nombre entier positif n et d'afficher son équivalent en octal.

Réponse :

```

#include <stdio.h>
#include <stdlib.h>
int main(){
    int n,ch,oct,i,j;
    printf("Donner un nombre entier >=0: ");
    scanf("%d",&n);
    if(n<0)printf("Erreur de saisi");
}

```

```

else{
    oct=0;i=0;
    while(n!=0){
        ch=n%8;
        n=n/8;
        for(j=1;j<=i;j++)
            ch=ch*10;
        oct=oct+ch;
        i=i+1;
    }
    printf("L'equivalent en octal est: %d",oct);
}
return 0;
}

```

Exercice 63 :

Écrivez un programme pour entrer un nombre et vérifiez si ce nombre est parfait ou non.

Un nombre est dit parfait s'il est égal à la somme de ses diviseurs positifs stricts.

Exemple: 28 est un nombre parfait. Les diviseurs appropriés de 28 sont 1, 2, 4, 7, 14.

Somme de ses diviseurs stricts = 1 + 2 + 4 + 7 + 14 = 28.

Par conséquent, 28 est un nombre parfait.

Réponse :

```

#include <stdio.h>
#include <stdlib.h>
int main(){
int n,i,s;
printf("Donner un nombre entier positif: ");
scanf("%d",&n);
if(n<=0)printf("Erreur de saisi");
else{
    s=0;
    for(i=1;i<n;i++){
        if(n%i==0)s=s+i;
    }
    if(s==n)printf("%d est un nombre parfait",n);
    else printf("%d n'est pas un nombre parfait",n);
}
return 0;
}

```

Exercice 64 :

Supposons que n est un entier dont l'écriture décimale comporte m chiffres. On dira que n est *narcissique* (nombre d' Armstrong) lorsque la somme des puissances m de chacun de ses chiffres est égale à n .

Exemple: $n=153$ est un nombre narcissique car $1^3+5^3+3^3=1+125+27=153$.

$n = 1634$ est un nombre narcissique car $1^4+6^4+3^4+4^4 = 1+ 1296+ 81+ 256=1634$.

Ecrire un programme permettant de lire un nombre entier positif et de déterminer s'il est narcissique.

Réponse :

```

#include <stdlib.h>
#include <math.h>
int main() {
    int n,nbChiffres,ch,n2,res,puiss,i;
    printf("Donner un nombre entier >0: ");
    scanf("%d",&n);
    nbChiffres=0;n2=n;
    while(n2!=0) {
        n2=n2/10;
        nbChiffres=nbChiffres+1;
    }
    n2=n;res=0;
    while(n2!=0) {
        ch=n2%10;
        n2=n2/10;
        puiss=1;
        for(i=1;i<=nbChiffres;i++)puiss=puiss*ch;
        res=res+puiss;
    }
    if(res==n)printf("%d est un nombre narcissique",n);
    else printf("%d n'est pas un nombre narcissique",n);
    return 0;
}

```

Exercice 65 :

Écrivez un programme permettant de saisir un nombre de l'utilisateur et vérifiez si le nombre est *Fort* ou non.

Un nombre fort est un nombre spécial dont la somme de la factorielle de ses chiffres est égale au nombre d'origine.

Par exemple: 145 est un nombre fort, puisque, $1! + 4! + 5! = 145$

Réponse :

```

#include <stdio.h>
int main() {
    int n,n2,s,f,ch,i;
    printf("Donner un nombre entier >0: ");
    scanf("%d",&n);
    if(n<=0)printf("Erreur de saisi");
    else{
        s=0;n2=n;
        while(n2!=0) {
            ch=n2%10;
            n2=n2/10;
            if(ch==0) f=1;
            else{
                f=1;
                for(i=1;i<=ch;i++) f=f*i;
            }
            s=s+f;
        }
        if(n==s)printf("%d est un nombre fort",n);
        else printf("%d n'est pas un nombre fort",n);
    }
}

```

```

    }
    return 0;
}

```

Exercice 66 :

Écrire un programme qui saisit un nombre entier positif non nul est qui vérifie si ce nombre est une puissance de trois ou pas. Si c'est le cas, vous devez afficher l'exposant qui va avec.

Réponse :

```

#include <stdio.h>
int main() {
    int n,a,p,i;
    printf("Donner un nombre entier >0: ");
    scanf("%d",&n);
    if(n<=0)printf("Erreur de saisi");
    else{
        a=0;p=1;
        for(i=1;i<=a;i++)
            p=p*3;
        while(p<n){
            a=a+1;p=p*3;
            for(i=1;i<=a;i++)
                p=p*3;
        }
        if(p==n)printf("%d est une puissance de 3, car 3^%d=%d ",n,a,n);
        else printf("%d n'est pas une puissance de 3",n);
    }
    return 0;
}

```

Commentaire [DZ1]: زائدة. يمكن إزالتها

Réponse 2:

```

#include<stdio.h>
#include<math.h>
int main() {
    int n;float r;
    printf("Donner un puissance de 3: ");
    scanf("%d",&n);
    r=n;
    while(r>3){
        r=r/3;
    }
    if(r==3)printf("%d est un multiple de 3",n);
    else printf("%d n'est pas un multiple de 3",n);
    return 0;
}

```

Exercice 67 :

Écrire un programme qui affiche la table de multiplication pour N variant de 1 à 9. L'affichage devra être comme suit:

x*y	1	2	3	4	5	6	7	8	9
1	1	2	3	4	5	6	7	8	9
2	2	4	6	8	10	12	14	16	18

```

3 | 3  6  9 12 15 18 21 24 27
4 | 4  8 12 16 20 24 28 32 36
5 | 5 10 15 20 25 30 35 40 45
6 | 6 12 18 24 30 36 42 48 54
7 | 7 14 21 28 35 42 49 56 63
8 | 8 16 24 32 40 48 56 64 72
9 | 9 18 27 36 45 54 63 72 81
10| 10 20 30 40 50 60 70 80 90
---|-----

```

Réponse :

```

#include <stdio.h>
#include <stdlib.h>
int main(){
    int i,j,r;
    printf("X*Y|\t");
    for(i=1;i<=8;i++)
        printf("%d\t",i);
    printf("9\n");
    printf("---|-----\n");
    for(i=1;i<=10;i++){
        if(i!=10)printf("%d |\t",i);
        else printf("%d |\t",i);
        for(j=1;j<=9;j++){
            r=i*j;
            if(j!=9)printf("%d\t",r);
            else printf("%d\n",r);
        }
    }
    printf("---|-----\n");
    return 0;
}

```

Exercice 68:

Ecrire un programme qui demande de saisir 6 entiers et qui affiche la valeur maximale saisie ainsi que son nombre d'occurrences dans la série.

Réponse :

```

#include<stdio.h>
int main() {
    #define n 6
    int i,v,max,nbOcc;
    printf("Donner la 1ere valeur: ");
    scanf("%d",&v);
    max=v;nbOcc=1;
    for(i=2;i<=n;i++){
        printf("Donner la %deme valeur: ",i);
        scanf("%d",&v);
        if(v==max)nbOcc=nbOcc+1;
        else if(v>max){
            max=v;
            nbOcc=1;
        }
    }
}

```

```

    }
}
printf("Le max est %d, il apparait %d fois dans la serie", max, nbOcc);
return 0;
}

```

Exercice 69:

On veut calculer $\sum_{i=a}^b \frac{i+k}{\cos(i-k)}$ or il risque d'y avoir des divisions par 0.

Ecrire un programme permettant de saisir la valeur de a, b, et k, et calcule et affiche la somme précédente en évitant les divisions par 0.

Réponse :

```

#include <stdio.h>
#include <math.h>
int main() {
    int a,b,k,i; float s;
    printf("Donner la valeur de a,b, et k: ");
    scanf("%d%d%d", &a, &b, &k);
    if(a>b) printf("erreur de saisi");
    else {
        s=0;
        for(i=a; i<=b; i++) {
            if(cos(i-k) != 0)
                s=s+(i+k)/(cos(i-k));
        }
        printf("La somme est %.3f", s);
    }
    return 0;
}

```

Exercice 70 :

Une personne dispose de 200000 DA de liquidités. Il souhaite placer ce montant sur un compte rémunéré qui lui rapportera des intérêts.

Ecrivez un programme qui calcule le montant dont pourra disposer cette personne selon le taux d'intérêt et la durée de placement.

Exemple : Montant initial : 200000 DA
 Durée de placement : 5 ans
 Taux de rémunération (en pourcentage) : 2,5
 Montant après 5 ans : 226282.16 DA

Réponse :

```

#include <stdio.h>
#include <stdlib.h>
int main() {
    float mont, taux; int nbAn, i;
    printf("Donner le montant initial: ");
    scanf("%f", &mont);
    printf("Donner la duree de placement (en annees): ");
    scanf("%d", &nbAn);
}

```

```

printf("Donner le taux d'interet (%%): ");
scanf("%f",&taux);
if(mont<=0 || nbAn<=0 || taux<0 || taux>100)printf("Erreur de saisi");
else{
    for(i=1;i<=nbAn;i++){
        mont=mont+mont*taux/100;
    }
    printf("Le montant apres %d ans est: %.2f DA",nbAn,mont);
}
return 0;
}

```

Exercice 71 :

La banque X vous accorde un prêt si la somme de vos intérêts dépasse 50000 DA.

L'intérêt est de 3.5 % par an.

Voici un exemple pour vous guider dans vos affaires financières:

Somme initiale placée: 700000 DA

1^{ère} année : intérêt = $(700000 \times 3.5)/100 = 24500$ DA somme = 724500 DA

2^{ème} année : intérêt = $(724500 \times 3.5)/100 = 25357.5$ DA somme = 749857.5 DA

...

On arrête quand intérêt > 50000 DA.

En suivant l'exemple précédent, écrire un programme C qui lit la somme d'argent placée initialement, puis détermine le nombre d'années nécessaires pour bénéficier d'un prêt. Le programme doit afficher le montant d'intérêt et la somme d'argents totale dans chaque année.

Réponse :

```

#include<stdio.h>
int main() {
    int nbAnnees;
    float somme, interet;
    interet=0;nbAnnees=0;
    printf("Donner la somme initiale: ");
    scanf("%f",&somme);
    printf("-----\n");
    printf("Annee\t\t Interet\t\tSomme\n");
    printf("-----\n");
    while(interet<50000) {
        nbAnnees=nbAnnees+1;
        interet=somme*3.5/100;
        somme=somme+interet;
        printf("Annee=%d \tinteret=%.2f\t\tsomme=%.2f\n",nbAnnees,interet,
somme );
    }
    printf("-----\n");
    return 0;
}

```

Exercice 72 :

Ecrire un programme pour calculer la somme des n premiers termes de la suite suivante :

$$U_n = \frac{4+2n}{3n}, \quad U_0 = 1.$$

Exemple : avec $n = 3$, $U_n = U_0 + U_1 + U_2 + U_3 = 1 + \frac{4+2 \times 1}{3 \times 1} + \frac{4+2 \times 2}{3 \times 2} + \frac{4+2 \times 3}{3 \times 3} = 1 + \frac{6}{3} + \frac{8}{6} + \frac{10}{9} = 5.444$

Réponse :

```
#include <stdio.h>
#include <stdlib.h>
int main(){
    int n,i;float un;
    printf("Donner la valeur de n: ");
    scanf("%d",&n);
    if(n<0)printf("Erreur de saisi");
    else{
        if(n==0)un=1;
        else{
            un=1;
            for(i=1;i<=n;i++)
                un=un+(float)(4+2*i)/(3*i);
        }
        printf("un=%.2f",un);
    }
    return 0;
}
```

Exercice 73 :

Calculez la racine carrée X d'un nombre réel positif A par approximations successives en utilisant la relation de récurrence suivante:

$$X_{j+1} = \frac{\left(X_j + \frac{A}{X_j}\right)}{2}, \quad X_1 = A$$

La précision du calcul J est à entrer par l'utilisateur.

Assurez-vous lors de l'introduction des données que la valeur pour A est un réel positif et que J est un entier naturel positif, plus petit que 50.

Réponse :

```
#include <stdio.h>
#include <stdlib.h>
int main(){
    float a,x;int j,i;
    do{
        printf("Donner un nombre reel positif A: ");
        scanf("%f",&a);
    }while(a<=0);
    do{
        printf("Donner le precision de calcul (entier): ");
        scanf("%d",&j);
    }while(j<=0 || j>=50);
    x=1;
    for(i=1;i<=j;i++){
        x=(x+a/x)/2;
    }
```

```

    }
    printf("la racine carree de %.2f est %f",a,x);
    return 0;
}

```

Exercice 74 :

Sachant que le développement du cos est le suivant :

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} + \dots + (-1)^i \frac{x^{2i}}{(2i)!} + \dots + (-1)^n \frac{x^{2n}}{(2n)!}$$

Ecrire un programme permettant de lire un nombre entier n , et un nombre réel x , puis calcule et affiche l'approximation de $\cos(x)$ au rang n .

Réponse :

```

#include <stdio.h>
#include <stdlib.h>
int main(){
    int n,i,j,f,signe;float xdegres,xrad,cos,p;
    printf("Donner l'angle x en degres: ");
    scanf("%f",&xdegres);
    printf("Donner un nmbre entier >= 0: ");
    scanf("%d",&n);
    if(n<0)printf("Erreur de saisi");
    else{
        xrad=3.14*xdegres/180;
        if(n==0)cos=1;
        else{
            cos=1;
            for(i=1;i<=n;i++){
                if(i%2==0)signe=1;
                else signe=-1;
                p=1;
                for(j=1;j<=2*i;j++)p=p*xrad;
                f=1;
                for(j=1;j<=2*i;j++)f=f*j;
                cos=cos+signe*p/f;
            }
            printf("cos(%.2f)=%.3f",xdegres,cos);
        }
    }
    return 0;
}

```

Exercice 75 :

Écrire un programme qui permette de connaître ses chances de gagner au tiercé, quarté, quinté et autres impôts volontaires. On demande à l'utilisateur le nombre de chevaux partants, et le nombre de chevaux joués. Les deux messages affichés devront être :

« Dans l'ordre : une chance sur X de gagner »

« Dans le désordre : une chance sur Y de gagner »

X et Y nous sont donnés par la formule suivante :

$$X = \frac{n!}{(n-p)!} \qquad Y = \frac{n!}{p!(n-p)!}$$

Tel que n est le nombre de chevaux partants et p le nombre de chevaux joués, alors :

Réponse :

```
#include <stdio.h>
#include <stdlib.h>
int main() {
    int n,p,i,f1,f2,f3;float x,y;
    printf("Donner le nombre de chevaux partants: ");
    scanf("%d",&n);
    printf("Donner le nombre de chevaux joués: ");
    scanf("%d",&p);
    if(n<=0 || p<=0 || n<p)printf("Erreur de saisi");
    else{
        f1=1;
        for(i=1;i<=n;i++) f1=f1*i;
        f2=1;
        for(i=1;i<=n-p;i++) f2=f2*i;
        f3=1;
        for(i=1;i<=p;i++) f3=f3*i;
        x=(float)f1/f2;
        y=(float)f1/(f2*f3);
        printf("Dans l'ordre: une chance sur %.0f de gagner\n",x);
        printf("Dans le desordre: une chance sur %.0f de gagner",y);
    }
    return 0;
}
```

Exercice 76 :

A) Une personne a acheté une nouvelle voiture. Il veut connaître à quel prix il pourra revendre sa voiture quelques années plus tard sachant que la valeur de la voiture diminue d'un taux fixe chaque année.

Ecrire un programme qui demande à l'utilisateur de saisir le prix de la voiture, le nombre d'années d'utilisation prévues, et le taux de diminution de prix. Le programme devra ensuite calculer le prix de la voiture au moment de la revente et affichera ce prix.

Exemple : Prix d'achat : 1300000 DA

Durée d'utilisation : 2 ans

Taux de diminution : 15%

Valeur de revente au bout de 2 ans : 939250 DA

B) La personne veut maintenant savoir combien lui aura coûté sa voiture au moment de la revente en tenant compte des dépenses de carburant.

En estimant que le litre de d'essence est stable et vaut 35 DA et que la personne roule 20000 km par an, et que la voiture consomme 6 litres pour 100 km, le coût dépensé pour l'essence sera de 42000 DA par an ($42000 = (20000 * 6 / 100) * 35$).

Pour simplifier le travail à faire, on considère que la valeur de 42000 DA par an est une constante.

Exemple : Prix d'achat : 1300000 DA

Durée d'utilisation : 2 ans

Coût de l'essence : $42000 * 2 = 84000$ DA

Coût de revient total : $1300000 + 84000 - 939250 = 444750$ DA

Budget mensuel à prévoir: $9622.50 / 24 = 18531.25$ DA

Ecrivez les modifications à apporter au programme précédent pour afficher le coût de revient de la voiture ainsi que le budget mensuel que la personne devra prévoir.

Réponse :

A)

```
#include <stdio.h>
#include <stdlib.h>
int main() {
    int nbAn,i;float mont,taux;
    printf("Donner le prix d'achat de la voiture: ");
    scanf("%f",&mont);
    printf("Donner le nombre d'annees d'utilisation: ");
    scanf("%d",&nbAn);
    printf("Donner le taux de diminution (%): ");
    scanf("%f",&taux);
    if(mont<=0 || nbAn<=0 || taux<0 || taux>100)printf("Erreur de saisi");
    else{
        for(i=1;i<=nbAn;i++)
            mont=mont-mont*taux/100;
        printf("La valeur de revente dans %d ans est: %.2f DA",nbAn,mont);
    }
    return 0;
}
```

B)

```
#include <stdio.h>
#include <stdlib.h>
int main() {
    int nbAn,i;float mont,rev,taux,ess,tot,budjMens;
    printf("Donner le prix d'achat de la voiture: ");
    scanf("%f",&mont);
    printf("Donner le nombre d'annees d'utilisation: ");
    scanf("%d",&nbAn);
    printf("Donner le taux de diminution (%): ");
    scanf("%f",&taux);
    if(mont<=0 || nbAn<=0 || taux<0 || taux>100)printf("Erreur de saisi");
    else{
        rev=mont;
        for(i=1;i<=nbAn;i++)
            rev=rev-rev*taux/100;
        ess=42000*nbAn;
        printf("Le cout de l'essence dans %d ans est: %.2f\n",nbAn,ess);
        tot=mont+ess-rev;
        printf("Le cout de revient total est: %.2f DA\n", tot);
        budjMens=tot/(nbAn*12);
        printf("Le budget mensuel a prévoir est: %.2f DA",budjMens);
    }
}
```

```
    }
    return 0;
}
```

Exercice 77:

Écrire un programme qui lit une somme d'argent puis une liste de dépenses, une par une, jusqu'à ce que la somme s'épuise où devient insuffisante. Afficher dans ce dernier cas le supplément d'argent nécessaire pour couvrir toutes les dépenses y compris la dernière.

Réponse :

```
#include <stdio.h>
#include <stdlib.h>
int main(){
    float somme,depense;
    do{
        printf("Donner une somme d'argent: ");
        scanf("%f",&somme);
        if(somme<=0)printf("Erreur,");
    }while(somme<=0);
    while(somme>0){
        do{
            printf("Donner une depense: ");
            scanf("%f",&depense);
            if(depense<=0)printf("Erreur,");
        }while(depense<=0);
        somme=somme-depense;
    }
    if(somme==0)printf("C'est tout ce que vous pouvez faire avec cette
somme");
    else printf("Il vous faudra %.2f DA de plus pour faire tout ceci",-
somme);
    return 0;
}
```

Exercice 78 :

Écrire un programme permettant de lire la suite des prix (en DA, entiers, et terminée par zéro) des achats d'un client. Calculer la somme qu'il doit, lire la somme qu'il paye, et déterminer le reste à rendre.

Réponse :

```
#include <stdio.h>
#include <stdlib.h>
int main(){
    int prix,somme,reste,pay;
    printf("Donner le 1er prix: ");
    scanf("%d",&prix);
    somme=0;
    while(prix!=0){
        if(prix>0)somme=somme+prix;
        else printf("Prix incorrect, ");
        printf("Donner un autre prix: ");
        scanf("%d",&prix);
    }
}
```

```

if(somme==0)printf("Vous n'avez rien a payer");
else {
    do{
        printf("Il vous doit: %d DA\n",somme);
        printf("Combien vous payez: ");
        scanf("%d",&pay);
        if(pay<0)printf("Erreur, ");
    }while(pay<somme);
    reste=pay-somme;
    printf("Le reste a rendre est: %d DA",reste);
}
return 0;
}

```

Exercice 79:

Un fermier fait l'élevage de montons et de dindons et, au moment de payer ses impôts, il déclare: j'ai dans mon élevage 36 têtes et 100 pattes !

Ecrire un programme pour déterminer le nombre de moutons et de dindons selon le schéma suivant (M: moutons, D: dindons, p: pattes)

Si M=36 alors D=0 donc $p=36 \times 4 = 144$ impossible

Si M=35 alors D=1 donc $p=35 \times 4 + 1 \times 2 = 142$ impossible

Si M=34 alors D=2 donc $p=34 \times 4 + 2 \times 2 = 140$ impossible

.....

jusqu'à ce que $p=100$.

Réponse :

```

#include<stdio.h>
int main() {
    int M,D,p,solution;
    M=36;
    D=0;
    p=M*4+D*2;
    solution = 1;// Solution=1 si le probleme a une solution et 0 sinon
    while(p!=100 && M>1) {
        M=M-1;
        D=D+1;
        p=M*4+D*2;
    }
    if(M==1 && p!=100) solution=0;
    if(solution==1) {
        printf("Le nombre de Moutons: %d\n",M);
        printf("Le nombre de Dindons: %d",D);
    }
    else printf("Ce probleme n'a pas de solution");
    return 0;
}

```

Exercice 80:

Écrivez un programme pour saisir un nombre de l'utilisateur et échangez le premier et le dernier chiffre du nombre donné.

Exemple: si le nombre saisi est 7891, après l'échange il devient 1897.

Réponse :

```
#include<stdio.h>
#include<math.h>
int main() {
    int n,prem,dern,n2,nbChiffres,puiss;
    printf("Donner un nombre entier >0: ");
    scanf("%d",&n);
    if(n<=0)printf("Erreur de saisi");
    else{
        n2=n;
        nbChiffres=0;
        prem=n%10;
        n=n/10;
        while(n>0){
            n=n/10;
            nbChiffres=nbChiffres+1;
        }
        puiss=round(pow(10,nbChiffres)); //La fonction round() arrondit un
nombre fractionnaire à l'entier le plus proche.
        dern=n2/puiss;
        n2=n2%puiss;
        n2=n2-prem+dern;
        n2=n2+prem*puiss;
        printf("Après l'echange: %d",n2);
    }
    return 0;
}
```

Exercice 81 :

Écrivez un programme qui demande à l'utilisateur de taper une phrase, puis qui affiche le nombre de « e » qu'il y a dans celle-ci. Une phrase se termine toujours par un point « . », un point d'exclamation « ! » ou un point d'interrogation « ? ». Pour effectuer cet exercice, il sera indispensable de lire la phrase dans sa totalité.

Par exemple: la phrase "Bonjour, comment allez-vous ?" contient 2 lettres "e".

Réponse :

```
#include <stdio.h>
int main(void){
    int compteur;char c;
    printf("Entrez une phrase se terminant par '.', '!' ou '?' : ");
    compteur = 0;
    do{
        scanf("%c", &c);
        if (c == 'e' || c == 'E')
            compteur++;
    } while (c != '.' && c != '!' && c != '?');
    if (compteur == 0)
        printf("Aucune lettre 'e' n'est reperee");
    else
        printf("%d lettres 'e' ont ete reperee", compteur);
}
```

```

    return 0;
}

```

Exercice 83 :

Le but dans cet exercice est d'écrire un programme qui permet de simuler le déplacement d'un robot (appelons le "*Crabby*") le long d'un axe. *Crabby* donc, peut se déplacer d'une case vers la gauche ou vers la droite.

Simplement, l'environnement dans lequel il évolue est fini et constitué de n cases alignées. n est donné par l'utilisateur.

Il va falloir écrire le programme qui simule le déplacement de *Crabby* en fonction de l'ordre qu'on lui donne. Au départ, le programme doit interroger l'utilisateur qui détermine le nombre de cases de l'environnement et la position initiale de *Crabby*. Attention, il faut confirmer que la position initiale est comprise entre 1 et n .

Le programme affiche les cases de l'environnement à l'aide de caractères. Le caractère "o" désignera une case vide et le caractère "X" la case où se trouve *Crabby*. Par exemple, si la position initiale est 7 dans un environnement composé de 8 cases, on affiche : oooooXo

L'utilisateur donne, ensuite, des ordres à *Crabby* en lui précisant le sens de déplacement ("G" : vers la gauche, "D" : vers la droite). Les lettres de déplacement doivent être en majuscule.

Il affichera, après chaque ordre, les cases de l'environnement comme dans le cas initial.

Par exemple : oooooXo \xrightarrow{G} oooooXoo \xrightarrow{D} oooooXo \xrightarrow{D} ooooooX \xrightarrow{D} ooooooX

Le programme s'arrête lorsque l'utilisateur saisi l'ordre "E".

Le programme établi doit prédire tous les cas possibles.

Réponse :

```

#include <stdio.h>
#include <stdlib.h>
int main(){
    int n,pos,i;char ordre;
    do{
        printf("Donner la taille de l'environnement: ");scanf("%d",&n);
    }while(n<=0);
    do{
        printf("Donner la position initial de Crabby: ");scanf("%d",&pos);
    }while(pos<=0 || pos>n);
    for(int i=1;i<pos;i++)printf("o");
    printf("X");
    for(int i=pos+1;i<=n;i++)printf("o");
    do{
        do{
            printf("\nDonner un ordre Crabby (G:gauche/D:Droite/E:Quiter):
");
            scanf("%c",&ordre); scanf("%c",&ordre);
        }while(ordre!='G' && ordre!='D' && ordre!='E');
        if(ordre=='G' && pos!=1)pos=pos-1;
        else if(ordre=='D' && pos!=n)pos=pos+1;
        for(int i=1;i<pos;i++)printf("o");
        printf("X");
        for(int i=pos+1;i<=n;i++)printf("o");
    }while(ordre!='E' && ordre!='e');
    return 0;
}

```

```
}
```

Exercice 70 :

Ecrire un programme permettant de lire une suite de nombres entiers non-nuls, et terminée par zéro, et qui détermine le nombre des valeurs positives et le nombre des valeurs négatives dans la suite.

Réponse :

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int n,nbPos,nbNeg;
    nbPos=0;nbNeg=0;
    do{
        printf("Donner un nombre entier: ");scanf("%d",&n);
        if(n>0)nbPos=nbPos+1;
        else if(n<0)nbNeg=nbNeg+1;
    }while(n!=0);
    printf("Le nombre de valeurs positives est: %d\n",nbPos);
    printf("Le nombre de valeurs negatives est: %d\n",nbNeg);
    return 0;
}
```

Solutions

Exercice 51 :

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