

**2 H - Sans documents ni calculatrice (sauf diagrammes de C)**  
 La clarté de vos réponses sera prise en compte. Ne pas écrire au crayon.  
**N'oubliez pas de commenter vos programmes.**

<b>Vous aurez besoin de 3 copies : une par partie</b>
---

---

### Partie 1 (6 points) → copie n°1

---

1/ Qu'affiche le programme ? (1,5 points)

```
void main()
{   int n1,n2,*p1,*p2 ;

    n1 = 9 ;
    p1= &n2 ;
    p2 = p1 ;
    *p2 = n1 % 5 ;
    printf("\n n1= %d n2= %d, p1 = %d, p2= %d", n1,n2,*p1,*p2) ;
    n1 = n1/4;
    n1*=n2+3 ;
    p2= &n1 ;
    n2++ ;
    printf("\nn1= %d n2= %d, p1 = %d, p2= %d", n1,n2,*p1,*p2) ;
    n1-- ;
    n2 = 20>n1>1 ;
    if (*p2%2)
        printf("\n partie alors n1=%d n2=%d, p1=%d, p2=%d \n ", n1,n2,*p1,*p2);
    else
        printf("\n partie sinon n1=%d n2=%d, p1=%d, p2=%d \n ", n1,n2,*p1,*p2);
}
```

2/ Soit la boucle do while suivante : (1,5 points)

```
int i ,n, f;

printf ("entrez 2 nombres (positifs ou nul) SVP : ");
scanf ("%d %d",&n,&i);
f=0;
do { n--;
    f+=i;
}
while (n!= 0 && n<10);
```

Décrire à l'aide d'une formule ce que fait cette boucle et précisez les hypothèses sur les données pour que cette formule soit valable.

Ecrire le code équivalent en utilisant une boucle for.

3/ Analyse de mot (3 points)

Ecrire un programme qui affiche le plus petit et le plus grand caractère (par ordre alphabétique) d'un mot entré au clavier ainsi que le nombre de fois où ces caractères ont été entrés. On supposera que tous les caractères sont en minuscules, qu'il n'y a pas d'accents et que le dernier caractère est un #.

Cependant, les mots peuvent contenir d'autres caractères (espaces, tirets, etc...) qui ne seront pas pris en compte pour l'analyse demandée.

On souhaite réaliser deux programmes affichant des propriétés "curieuses" sur les nombres.

1/ Une somme de nombres impairs consécutifs est égale à un carré :

$$1 + 3 = 4 = 2^2$$

$$1 + 3 + 5 = 9 = 3^2$$

$$1 + 3 + 5 + 7 = 16 = 4^2$$

...

Ecrire un programme qui demande à l'utilisateur de saisir "n" (correspondant à 4 dans l'exemple précédent) et qui affiche le résultat suivant (exemple pour n=6) :

Entrez n : 6

Somme des nombres impairs consécutifs :

$$1 + 3 = 4$$

$$1 + 3 + 5 = 9$$

$$1 + 3 + 5 + 7 = 16$$

$$1 + 3 + 5 + 7 + 9 = 25$$

$$1 + 3 + 5 + 7 + 9 + 11 = 36$$

2/ Addition surprenante

$$1 + 2 = 3$$

$$4 + 5 + 6 = 7 + 8$$

$$9 + 10 + 11 + 12 = 13 + 14 + 15$$

$$16 + 17 + 18 + 19 + 20 = 21 + 22 + 23 + 24$$

...

Remarques :

- Chaque ligne commence par les carrés de 1, 2, 3, 4 ... soit "i" : 1, 4, 9, 16...

- Pour la "i"-ème ligne, il y a "i" termes à ajouter au carré de "i" sur l'expression de gauche, la somme sur la partie droite comprend "i" termes.

Ecrire un programme qui demande à l'utilisateur de saisir "n" correspondant au nombre de lignes et qui affiche le résultat suivant (exemple pour n=6) :

Entrez n : 6

Addition surprenante :

$$1 + 2 = 3$$

$$4 + 5 + 6 = 7 + 8$$

$$9 + 10 + 11 + 12 = 13 + 14 + 15$$

$$16 + 17 + 18 + 19 + 20 = 21 + 22 + 23 + 24$$

$$25 + 26 + 27 + 28 + 29 + 30 = 31 + 32 + 33 + 34 + 35$$

$$36 + 37 + 38 + 39 + 40 + 41 + 42 = 43 + 44 + 45 + 46 + 47 + 48$$

Ecrire un programme qui demande à un utilisateur la date du jour (jour, mois, année) et la date de son anniversaire (jour et mois [pas l'année]). Le programme calculera alors le nombre de jours restant avant le prochain anniversaire de l'utilisateur.

On rappelle que dans une année non bissextile, les mois de janvier, mars, mai, juillet, août, octobre, décembre comptent 31 jours, que les mois d'avril, juin, septembre, novembre comptent 30 jours, et que le mois de février compte 28 jours.

On rappelle que lors d'une année bissextile il y a 29 jours au mois de février au lieu de 28. Il y a une année bissextile une fois tous les 4 ans (pas exactement, mais on utilisera cette formule simplifiée).