

NB : LA QUALITÉ DE LA RÉDACTION ET LA RIGUEUR DU RAISONNEMENT SERONT PRISES EN COMPTE DANS LA NOTATION

EXERCICE I

1) *Que fait le programme suivant :*

```
1 #include <stdio.h>
2 main() {
3     long unknown;
4     int k;
5     int max;
6     int n;
7
8     scanf("%d", &max);
9     unknown = 0;
10    n = 0;
11    k = 1;
12    while (k <= max)
13    {
14        unknown = unknown + k;
15        n++;
16        k += 2;
17    }
18    printf(" %ld\n", unknown);
19 }
```

Prog 1

2) Expliquer le programme suivant :

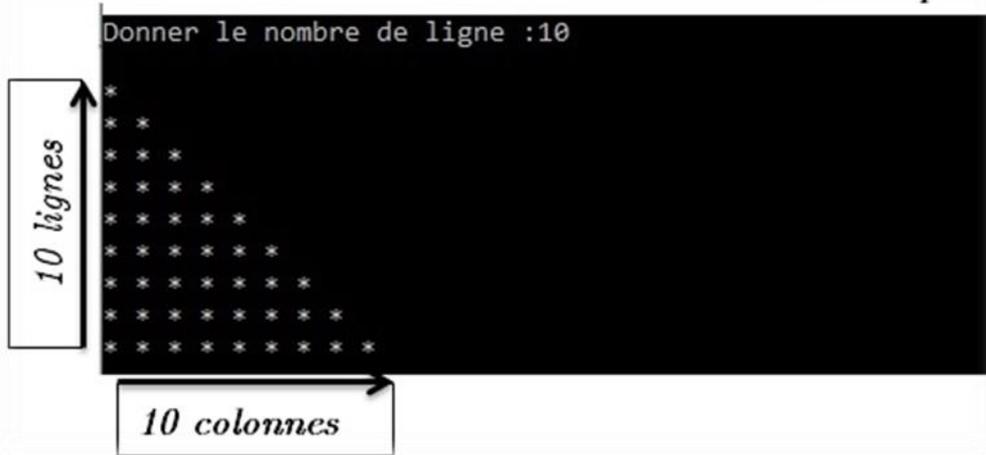
Prog 2

```
1  #include <stdio.h>
2  main()
3  {
4
5  long unknown = 122;
6  int z = 0;
7
8  while (!(unknown % 2)) {
9  scanf("%ld", &unknown);
10 z++;
11 }
12
13 printf("\n valeur --> %d .\n", z-1);
14 }
```

EXERCICE II

Ecrire un programme qui permet d'afficher un triangle rectangle formé d'étoiles de N lignes (N est fourni au clavier).

Exemple



2) Dédire le programme du triangle isocèle suivant :



EXERCICE III

Ecrire un programme qui lit un entier A et un tableau T de type `int` et qui permet de calculer le nombre des entiers supérieurs à A appartenant à T en utilisant le « formalisme pointeur »

Exemple : `Tab : [15 12 56 8 17 74]` `A=14` \rightarrow `4`

EXERCICE IV

Écrire un programme qui estime la valeur du **nombre de NEPER** e en utilisant la formule :

$$e = \sum_{i=0}^n \frac{1}{i!} = \frac{1}{0!} + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \dots + \frac{1}{n!}$$

```
Donner un entier n : 10
```

```
< i= 1 ---> e= 2.000000 >
```

```
< i= 2 ---> e= 2.500000 >
```

```
< i= 3 ---> e= 2.666667 >
```

```
< i= 4 ---> e= 2.708333 >
```

```
< i= 5 ---> e= 2.716667 >
```

```
< i= 6 ---> e= 2.718056 >
```

```
< i= 7 ---> e= 2.718254 >
```

```
< i= 8 ---> e= 2.718279 >
```

```
< i= 9 ---> e= 2.718282 >
```

```
< i= 10 ---> e= 2.718282 >
```