

Module : Algorithmique & Programmation1

Correction contrôle 2018_2019

/ Exercice1 contrôle 2018-2019*

*Ecrire un **algorithme** qui lit deux entiers $J > 1$ et $N > 1$ et qui calcule le produit suivant :*

$$P = \prod_{I=1}^N \frac{(I)^J}{PGCD(I,J)}$$

PGCD(I,J) désigne plus grand commun diviseur des deux entiers I et J.

**/*

//Déclaration

I,J,N,P=1,PGCD,K,IPJ,L :entier

//préparation du traitement

//lecture de J et N

Répéter

{ecris('donner deux entiers J et N > 1\n')}

Lis(J,N)

}

Tant que ((J<=1)|| (N<=1))

//traitement

Pour I allant de 1 jusqu'à N par pas de 1

{//calcul de IPJ=F'

IPJ=1

Pour K allant de 1 jusqu'à J par pas de 1

*IPJ=IPJ*K*

//calcul de PGCD(I,J)

Pour K allant de 1 jusqu'à I par pas de 1

Si(I%K==0)

Pour L allant de 1 jusqu'à J par pas de 1

Si((J%L==0)&&(L==K))

PGCD=K

*P=(P*IPJ)/PGCD*

}

//Edition des résultats

Ecris('P=',P)

}

*/*Exercice 2_1 contrôle 2018-2019*

Ecrire un programme qui lit un entier N impair et un entier NES et qui affiche les deux motifs suivants espacés sur la première et la dernière ligne par NES espaces. N est le nombre d'étoiles sur la première et la dernière ligne des deux motifs.

```
*****      *****
*      *      ***
*      *      *
*      *      ***
*****      *****
```

**/*

#include<stdio.h>

#include<stdlib.h>

int main()

{

//declaration

int N,I,NES,J,K=1;

//lecture de N

do

{printf("donner un entier N impair>=3\n");

scanf("%d",&N);

}

while ((N%2==0)|| (N<3));

//lecture de NES

do

{printf("donner un entier NES>=2\n");

scanf("%d",&NES);

}

while (NES<2);

//traitement

for(I=1;I<=N;I++)

{if((I==1)|| (I==N))

{for(J=1;J<=N;J++)

printf("");

for(J=1;J<=NES;J++)

printf(" ");

for(J=1;J<=N;J++)

printf("");

printf("\n");

}

if((I>1)&&(I<=N/2 +1))

```
{printf("*");
```

```
for(J=1;J<=N-2;J++)
printf(" ");printf("*");
for(J=1;J<=NES;J++)
printf(" ");
for(J=1;J<=I-1;J++)
printf(" ");
for(J=1;J<=N-(I-1)*2;J++)
printf("*");printf("\n");
}
if((I>N/2 +1)&&(I<N))
{K=K+2;printf("*");
for(J=1;J<=N-2;J++)
printf(" ");printf("*");
for(J=1;J<=NES;J++)
printf(" ");
for(J=1;J<=N-I;J++)
printf(" ");
for(J=1;J<=K;J++)
printf("*");printf("\n");
}
}
```

*/*Exercice 2_2 contrôle 2018-2019*

Ecrire un programme qui affiche un tapis comportant les motifs de l'exercice 2_1. Chaque rangée horizontale du tapis commence par le premier motif et se termine par le deuxième.

N nombre d' sur la première et la dernière ligne de chaque motif*

NH nombre de fois que les deux motifs sont répétés en horizontale

NV nombre de rangées formées les motifs en verticale

NES nombre d'espaces entre la première(resp dernière) ligne du premier motif et la première(resp dernière) ligne du deuxième motif

NL nombre de lignes vides séparant deux rangées en verticale

```
*/
#include<stdio.h>
#include<stdlib.h>
int main()
{
//declaration
int N,I,NES,J,NH,NL,NV,LV,LH;
//lecture de N
do
{printf("donner un entier N impair>=3\n");
scanf("%d",&N);
}while ((N%2==0)|| (N<3));
//lecture de NH,NL,NV
do
{printf("donner trois entiers NH,NL,NV impair>=3\n");
scanf("%d%d%d",&NH,&NL,&NV);
}while ((NH<=1)|| (NL<=1)|| (NV<=1));
//lecture de NES
do
{printf("donner un entier NES>=2\n");
scanf("%d",&NES);
}while (NES<2);
//traitement
for(LV=1;LV<=NV;LV++)
{for(I=1;I<=N;I++)
{for(LH=1;LH<=NH;LH++)
{
if((I==1)|| (I==N))
{for(J=1;J<=N;J++)
printf("*");
for(J=1;J<=NES;J++)
printf(" ");
for(J=1;J<=N;J++)
printf("*");
for(J=1;J<=NES;J++)
printf(" ");
}
}
if((I>1)&&(I<=N/2 +1))
{printf("*");
for(J=1;J<=N-2;J++)
printf(" ");printf("*");
for(J=1;J<=NES;J++)
```

```

printf(" ");
for(J=1;J<=I-1;J++)
printf(" ");
for(J=1;J<=N-(I-1)*2;J++)
printf("*");
for(J=1;J<=NES+I-1;J++)
printf(" ");
}
if((I>N/2 +1)&&(I<N))
{printf("*");
for(J=1;J<=N-2;J++)
printf(" ");printf("*");
for(J=1;J<=NES;J++)
printf(" ");
for(J=1;J<=N-I;J++)
printf(" ");
for(J=1;J<=2*I-N;J++)
printf("*");
for(J=1;J<=NES;J++)
printf(" ");
for(J=1;J<=N-I;J++)
printf(" ");
}
} //LH
printf("\n");
} //I
for(J=1;J<=NL;J++)
printf("\n");
} //LV
}

```

*/*Exercice 3 contrôle 2018-2019*

Ecrire **un programme** qui lit un entier $N > 1$ et qui permet:

- 1) Si N est premier, on cherche les nombres premiers $< N$
- 2) Si N n'est pas premier, on cherche le premier nombre premier atteint en incrémentant successivement N

```

*/
#include<stdio.h>
#include<stdlib.h>
int main()
{
//declaration
int N,I,B=0,K,L,J,D,E;
//lecture de N
//on sait que 1, 2, 3 on peut donc demander un
N>3

```

```

do
{printf("donner un entier N > 3\n");
scanf("%d",&N);
}
while (N<=3);
for(I=2;I<=N/2;I++)
if(N%I==0) B++;
if(B==0)
{printf("%d est un nombre
premier\n",N);
printf("les nombres premiers<%d
sont:\n",N);
J=N-1;
while(J>1)
{
for(K=J;K>=1;K--)
{D=0; for(L=2; L<=K/2; L++)
if(K%L==0) D++;
if(D==0) {printf("J=%d\n",J);}
J--;
}
}
}
else
{printf("%d n est pas premier\n",N);
printf("le premier nombre premier
atteint\n");
printf("en incrémentant successivement N
de 1 est N=");
EE: {N++; E=0;
for(I=2;I<=N/2;I++)
if(N%I==0) E++;
} while(E!=0) goto EE;
printf("%d\n",N);
}
printf("FIN\n");
}

```