Serie num 4 :

Pointeurs et fonctions

exercice 1 :

	A	B	C	P1	P2
	1	2	3	&A	&
*P1 = (*P2)++ -> { *P1 = *P2 *P2++	3	2	4	&A	&C
P1 = P2	3	2	4	&C	&C
P2 = &B	3	2	4	&C	&B
*P1 = *P2 => *P1 = *P1 - *P2 = 4 - 2 = 2	3	2	2	&C	&B
++*P2	3	3	2	&C	&B
*P1 *= *P2 => *P1 = (*P1)*(*P2) = 2*3 = 6	3	3	6	&C	&B
A = ++*P2 * *P1 => { *P2++ => *P2 = 4 A = 4*6 = 24	24	4	6	&C	&B

	P1 = &A	24	4	6	&A	&B
	*P2 = *P1 /= *P2					
=>	{ *P1 = *P1 / *P2 = 6	6	6	6	&A	&B
	{ *P2 = 6					

exercice 2 :

```
# include < stdio. h>
# include < stdlib. h>
# include < stddef. h>
main ()
{ int n, *p, x, T, *p1, *p2;

  do { printf (" Entrez la taille du tableau ");
       scanf ("%d", &n); } while ( n <0);

 T = (int *) malloc ( n* sizeof (int));
 if (T == NULL)
    { printf (" pas assez de mémoire");
      system ("PAUSE");
    }
for ( p = T; p < T + n ; p + + )
    { printf (" Entrez T[%d] \n", p - T);
      scanf ("%d", p);
    }
p1 = T;
p2 = T + n - 1
```

```
while ( p1 < p2 ) { x = * p1;
                    * p1 = * p2 ;
                    * p2 = x;
                      1 + + ;
                    p2 - - ;
                  }
}
```

exercice 3 :

```
# include < stdio. h >
# include < stdlib. h >
# include < stddef. h >
main ( )
 { int n , * p, T , k
  do { printf (" Entrez la taille du tableau ");
       scanf ("%d, & n) ; } while ( n <0) ;

T = (int *) malloc ( n* sizeof (int)) ;
if ( T = = NULL)
  { printf (" pas assez de mémoire ") ;
    system (" PAUSE"); }
for ( p = T ; p < T + n ; p + + )
   { printf (" Entrez T [%d] \n ", p - T ) ;
     scanf ("%d", p ) ; }
k = T ;
for ( p = T ; p < T + n ; p + + )
   if ( * p ! = 0 )
      {* k = * p ;
         k ++ ; }
}
```

deuxième méthode

```
for (p = T ; p < T + n ; p++)
    if (*p == 0)
    { for (k = p; k < T + n - 1 ; k++)
            *k = *(k++);
      p--;
      n--;
    }
```

exercice 5 :

l'affichage :

B : dans fct, n = 10, p = 5, q = 20

A : dans main, n = 20, p = 5, q = 2

C : dans f, n = 10, p = 20, q = 2

exercice 6 :

$$C_n^k = \frac{n!}{k!\,(n-k)!}$$

écrivez une fonction qui permet de calculer les coeff binomiaux C_n^k :

```
int fct_coeff_binom (int n, int k)
{ int i, nF = 1, kF = 1, sF = 1, s, coeff;
   s = n - k;
   // calcul de n!
   for (i = 2; i <= n; i++)
      nF *= i;
   // calcul de k!
   for (i = 2; i <= k; i++)
      kF *= i;
   // calcul de (n-k)!
   for (i = 2; i <= s; i++)
      sF *= i;

   coeff = nF / kF * sF;
   return coeff;
}
```

programme qui calcul C_n^k :

```
# include <stdio.h>
# include <stdlib.h>
int fct_fact (int);
int fct_coeff_binom (int, int);
main()
{ int n, k, R;
  printf ("Entrez n et k : ");
  scanf ("%d%d", &n, &k);
  R = fct_coeff_binom (n, k);
  printf ("C(%d,%d) = %d \n", n, k, R);
  system ("PAUSE");
}
int fct_fact (int n)
{ int i, F = 1;
  for (i = 2; i <= n; i++) F *= i;
  return F;
}
int fct_coeff_binom (int n, int k)
{ int coeff, t;
  t = n - k;
  coeff = fct_fact (n) / (fct_fact (k) * fct_fact (t));
  return coeff;
}
```