Université Sidi Mohamed Ben Abdellah Faculté des Sciences et Techniques Département d'Informatique

Série N°3

TD Algorithmique & Programmation

Exercice 1

Définir les fonctions suivantes :

• Triangulaire(int n): qui retourne 1 si l'entier n est triangulaire et 0 sinon.

Un nombre N est dit triangulaire s'il existe un k > 0 tel que : N = 1 + 2 + ... + k

Exemple: 15 est un nombre triangulaire car 15 = 1+2+3+4+5

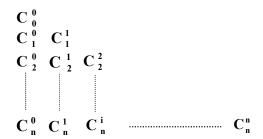
• **Tetraedrique(int n)** : qui retourne p si l'entier n est la somme des p premiers entiers triangulaires (on dit que n est un tétraedrique d'ordre p) et 0 sinon.

Exercice 2

Le nombre de combinaisons de i dans n est défini par :

$$C_n^i = n!/(i!*(n-i)!)$$

- 1. Ecrire une fonction qui calcule le nombre de combinaisons de i dans n
- 2. Le triangle de Pascal est défini par :



Ecrire une fonction qui édite le triangle de Pascal en respectant la forme d'affichage donnée

MIP2 1/3 2024-2025

Exercice 3

Donner le résultat d'exécution du programme suivant :

```
#include <stdio.h>
                              int reculer3(int *x){
                                                                          reculer2(y);
                                 *_{X}=3:
void avancer1(int x ){
                                                                          printf("\n y=\%d",y);
   x+=1;
                                 return *x;
                                                                          y=avancer2(&x);
                              }
                                                                          printf("n y=%d",y);
}
int avancer2(int *x){
                                                                          printf("\n x=\%d",x);
                              main(){
   *_{x}+=2;
                                   int x=6, y=1;
                                                                          y=reculer3(&x);
   return *x;
                                   avancer1(x);
                                                                          printf("\n y=\%d",y);
                                   printf("\n x=\%d",x);
                                                                          printf("\n x=\%d",x);
void reculer2(int x){
                                                                          x=reculer3(&y);
                                   x=avancer2(&y);
                                   printf("n y=%d",y);
  x=2:
                                                                          printf("\n x=\%d",x);
                                                                          printf("\n y=%d",y);
}
                                   printf("\n x=\%d",x);
```

Exercice 4

- 1. Ecrire une fonction *Max_Min* à 3 arguments qui calcule le maximum et le minimum d'une série de n entiers saisis au clavier. Les variables relatives au minimum et au maximum font partie des arguments de la fonction : void *Max_Min*(int, int)
- 2. Ecrire un programme qui fait appel à la fonction *Max Min* avec trois façons différentes.

```
int main()
                     int main()
                                           int main()
                                            int *n, *min, *max;
 int n, min, max
                      int n, *min, *max;
//contrôle de saisie sur n
                                             ......
                      //contrôle de saisie sur n
                                            //contrôle de saisie sur n
   .....
Max Min(..., ..., ...);
                      printf("\nLe max = \%d ", ....);
                                             printf("\nLe min = %d\n",....);
                      ......
                                             Max Min(..., ..., ....)
                                            Max Min(..., ..., ....)
                      printf("\nLe max = \%d ", ....);
                                            printf("\nLe max = \%d ", ....);
                      printf("\nLe min = %d\n",....);
                                            printf("\nLe min = \%d\n",....);
                      ......
                                             .....
```

Université Sidi Mohamed Ben Abdellah Faculté des Sciences et Techniques Département d'Informatique

Exercice 5

La forme récursive du nombre de combinaisons de i dans n est définie par :

$$C_n^i = C_{n-1}^i + C_{n-1}^{i-1}$$

Ecrire une fonction récursive qui calcule le nombre de combinaisons de i dans n

Exercice 6

Etant donnée la suite V_n définie par :

$$\begin{cases} V_0 = 2 \\ V_n = V_{n-1} + 2 & si \ n \ pair \\ V_n = V_{n-1} + 4 & si \ n \ impair \end{cases}$$

- 1. Ecrire une fonction récursive qui permet de calculer le $\mathbf{n}^{\text{ième}}$ terme de la suite V_n .
- **2.** Ecrire une fonction qui permet de calculer la somme des **N** premiers termes de la suite V_n .
- **3.** Ecrire un programme qui permet de saisir un entier N > 0, puis calcule la somme des N premiers termes de la suite V_n .

MIP2 3/3 2024-2025