```
int fct (int n) {
  switch (n) {
  case 0: return 1;
  case 1: return 2;
  case 2: return 6;
  default:
    return fct(n-1) + fct(n-2) - fct (n-3);
  }
}
```

TABLE 1 – Fonction fct

Programmation Impérative I

21/2/2018

Nom _____
Prénom _____
Login _____
mail _____

Les questions sont à traiter dans l'ordre qui vous plaira.
Ce sujet doit être remis avec votre copie, une version imprimable sera donnée sur la page du cours pour que vous puissiez retravailler les questions.
Quand vous simulez le fonctionnement de l'ordinateur lors de l'exécution d'une fonction, vous devez donner le contenu de la mémoire pour chacune des variables utilisées et ceci à chaque instant, comme on l'a fait en cours.

A) Récursions

1. Soit la fonction `fct`, cf. Table 1, donnez tous les appels récursifs et résultats obtenus lorsqu'on demande fct(6) ;
2. Faire une fonction qui renvoie la somme des entiers pairs inférieurs à un n donné.
3. Écrire la fonction récursive puiss qui renvoie la puissance `p` d'un nombre à virgule `x`.
   Notez que puiss (-3,2.0)==0.125
4. Écrire une fonction qui prend un `float` comme paramètre et renvoie sa valeur absolue.
5. Vous trouverez à la Table 2 la fonction `math` écrivez-la sous forme récursive.

```
float math (float b, float eps) {
  float x;
  x = 1.0;
  while (absol (x * x - b) > eps) {
    x = (x * x + b) / (2.0 * x);
  }
  return x;
}
```

TABLE 2 – Fonction fct

```
typedef double vecd_t [200];

void remplir (int nb, vecd_t v) {
  double x, y;
  int i;
  x = 10.0;
  y = 0.5;
  for (i = 0; i < nb; i++) {
    v[i] = x;
    x += y;
    if (x > 11.2) {
      y = 0.05 - y;
    }
    if (x < 8.9) {
      y = - y - 0.04;
    }
  }
}
```

TABLE 3 – Remplir un vecteur

B) Itérations

1. Écrire une fonction itérative qui fait le produit des entiers positifs impairs inférieurs à son argument.
   fct(8)=105
2. Écrire une fonction itérative qui calcule la somme des puissances n des premiers entiers jusqu'au rang n.
   som(4)=354
3. Vous avez vu la fonction `fct`, cf. Table 1, donnez une forme **itérative**, avec un while, de cette fonction.

C) Vecteurs

Soient la fonction `remplir` et le nouveau type `vecd_t`, donnés dans la table 3.

```
struct vec {
  int nb;
  float v [100];
} ;
typedef struct vec vec_t;
```

TABLE 4 – Structure de vecteur

1. Que se passe-t-il lors de l'exécution de la fonction `remplir` avec comme arguments un vecteur et la valeur 10 ?
   Utilisez le tableau ci-dessous pour noter les évolutions des variables.

nb	10
x	
y	
i	

v	0	1	2	3	4	5	6	7	8	9	10	11	12

2. Écrivez une fonction qui renvoie le plus grand élément d'un vecteur de type `vecd_t`.
3. Écrivez une fonction qui renvoie le second plus petit élément d'un vecteur de type `vecd_t`.
4. Écrivez une fonction qui renvoie la somme des éléments supérieurs à 1.0 du vecteur.
5. Écrivez une fonction qui, dans un tel vecteur, remplace chacune des valeurs par son carré.
6. Soient un vecteur de type `vecd_t`, que l'on suppose rangé dans l'ordre décroissant, son nombre d'éléments nb et une valeur x, écrivez une fonction qui insère la valeur x à sa place dans le vecteur de façon à ce que le vecteur reste trié dans l'ordre décroissant.
7. Utilisez la fonction précédente pour créer une fonction permettant de trier un tel vecteur.

D) Structures

Vous trouverez dans la table 4 une structure permettant de travailler avec un vecteur et dans la table 5 les valeurs qui se trouvent dans son champ v en supposant que son champ nb contient 10.

0	1	2	3	4	5	6	7	8	9
-1.0	1.2	-0.5	-0.3	0.6	0.8	1.1	-0.9	0.2	0.1

TABLE 5 – Valeurs du vecteur

```
vec_t change (vec_t w) {
  int im, i;
  im = 0;
  for (i = 1; i < w.nb; i++) {
    if (w.v[im] < w.v[i])
      im = i;
  }
  i--;
  w.v[im] += w.v[i];
  w.v[i] = w.v[im] - w.v[i];
  w.v[im] = w.v[im] - w.v[i];
  return w;
}
```

TABLE 6 – Utilisation du vecteur

1. Faire une fonction qui affiche les valeurs présentes dans cette structure.
2. Soit la fonction `change`, cf. Table 6, donnez toutes les étapes de son fonctionnement.
3. Donnez une fonction qui fait la sommes des valeurs positives du vecteur.
4. Soit δ_i l'écart entre la valeur en i et la valeur en i - 1
   $\delta_i = v.v[i] - v.v[i-1]$.
   Faire la somme de tous les δ_i positifs.
5. Donnez une fonction d'un tel vecteur et d'une valeur x qui renvoie 0 si $x \in v$ et 1 sinon.