

Examen de Programmation I

Exercice 1 (3 pts)

Soit un pointeur p qui pointe vers un tableau A comme suit :
int n = 5, *p, q, A[] = {12,23,34,45,56,67,78,89,90} ;
float x ;
p = A;

Quelles valeurs ou adresses de A fournissent les expressions suivantes :

Expression	p	q	n	x
1. q = n < *p ;				
2. q = p + n				
3. q = *p % n + *q > A[4] ;				
4. x = (float)(*p)/n ;				
5. x = n * (*p > n ? n++ : q) ;				
6. *q = x++ <= *(p+2) p++ != A+1 ;				
7. q = &A[3] - p + q				
8. x = !((++*p % n) ** p++)				

Exercice 2 (4 pts)

Ecrire un programme C qui demande un nombre entier de départ, et qui affiche les diviseurs de cet entier. Par exemple, si l'on entre 20, le programme affiche 2 4 5 et 10

Exercice 3 (6 pts)

Ecrire un programme permettant de tester une fonction dont le prototype est le suivant :

char * prefixecommun(char * chaine1, char * chaine2)

qui compare lettre à lettre deux chaînes de caractères et renvoie le plus grand préfixe commun. Cette fonction doit renvoyer le pointeur NULL si les deux chaînes n'ont aucun préfixe en commun.

Remarque : On pourra utiliser les fonctions de la bibliothèque string.h (gets, puts et strlen)

Exercice 4 (7 pts)

L'objectif de cet exercice est de développer un filtre non causal, c'est-à-dire une fonction qui lisse un signal en utilisant une fenêtre glissante pour moyenner les valeurs (figure ci-dessous). Pour les premières et dernières valeurs, seules les valeurs dans la fenêtre sont prises en compte.

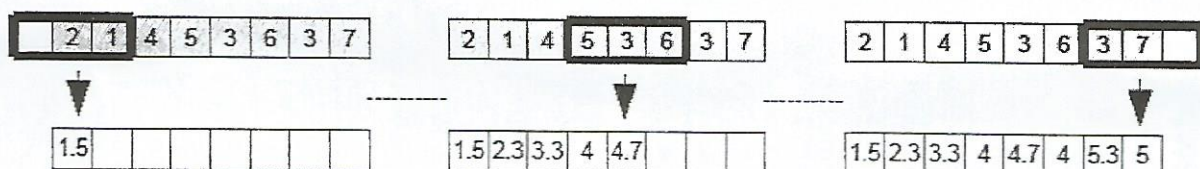


Figure – Lissage d'un signal avec une fenêtre de taille 3

- 1) Ecrire la définition de la fonction réalisant ce filtrage pour une fenêtre quelconque de taille N
- 2) Ecrire un programme permettant de tester cette fonction sur un signal aléatoire