

Problème	Algorithmie	C
Demander un nombre jusqu'à ce qu'il soit un multiple de 3	Entier nombre obtenir nombre tant que ((nombre % 3) ≠ 0) afficher nombre "pas multiple de 3" obtenir nombre fintantque afficher nombre "multiple de 3"	int nombre; scanf("%d", &nombre); while ((nombre % 3) != 0) { printf("%d pas multiple de 3\n", nombre); scanf("%d", &nombre); } printf("%d mult de 3\n", nombre);
Demander un nombre. Chercher le plus petit nombre carré qui soit strictement plus grand que ce nombre	Entier nombre, rac, solution obtenir nombre rac ← 0 tant que ((rac x rac) < nombre) rac ← rac + 1 fintantque solution ← rac x rac afficher solution	int nombre, rac, solution; scanf("%d", &nombre); rac ← 0 while ((rac * rac) < nombre) { rac = rac + 1; } solution = rac * rac; printf("solution : %d", solution);
Obtenir en nombre strictement positif (en le vérifiant)	Entier n faire obtenir n tant que (n < 1)	int n do { printf("Entrez un nombre >0 :"); scanf("%d", &n); } while (n < 1);
Demander un nombre. Afficher tous les nombres pairs inférieurs à celui-ci	Entier n, i obtenir n pour i de 2 à n par pas de 2 afficher i finpour	int n, i; scanf("%d", &n); for (i=2 ; i<= n ; i= i+2) { printf("%d\n", i); }

Méthodes

<pre> int plusGrandDiviseur(int a) { int div; div = a; do { div = div - 1; } while ((a % div) != 0); return div; } int ontMemePlusGdDiv(int a, int b) { int plusGdDivA = plusGrandDiviseur(a); int plusGdDivB = plusGrandDiviseur(b); return (plusGdDivA == plusGdDivB); } int main() { int c; c = plusGrandDiviseur(25); if (ontMemePlusGdDiv(49, 35) && (c == 5)) { printf("C'est bien ca !\n"); } system("pause"); return 0; } </pre>	<pre> float var; void doubleVar() { var = 2 * var; } float calcPuissVar(int puiss) { int i; float res; res = 1.0; for (i=0 ; i<puiss ; i=i+1) { res = res * var; } return res; } int main() { var = 3.2; doubleVar(); var = calcPuissVar(4); printf("var = %f", var); //1677,.. system("pause"); return 0; } </pre>
--	---

