

Introduction à la Programmation

Examen partiel sur machine

G1: monasse(at)imagine.enpc.fr G2: nicolas.audebert(at)onera.fr
G3: alexandre.boulch(at)onera.fr G4: antoine.recanati(at)inria.fr
G5: bourkia(at)imagine.enpc.fr G6: fadhela.kerdjoudj(at)u-pem.fr

02/11/16

1 Énoncé

1.1 Représentation graphique des décimales d'un nombre

On représente un nombre écrit en écriture décimale $a_0, a_1 a_2 a_3 a_4 \dots$ sous la forme de disques tangents ou imbriqués comme dans la figure 1 : trois disques pour la première décimale $a_0 = 3$, un disque dans le premier de ceux-ci pour $a_1 = 1$, quatre dans le deuxième disque pour $a_2 = 4$, un dans le troisième pour $a_3 = 1$, etc. Les disques sont mis dans une file d'attente après affichage et chaque nouvelle décimale s'affiche dans le premier de la file d'attente. Les cas particuliers sont (1) si $a_i = 1$ on affiche bien un seul disque mais avec un rayon légèrement plus petit, (2) si $a_i = 0$ on affiche 10 disques. Les couleurs sont arbitraires.

Il est plus important de livrer un code clair (commenté et indenté) et qui compile sans warning, même s'il ne répond pas à toutes les questions. Pour cela, vérifiez à chaque étape que votre programme compile et se lance correctement.

Créez un projet Imagine++ avec un fichier contenant le main. Prenez le CMakeLists.txt d'un projet existant et adaptez-le.

1.1.1 Extraction des décimales

1. Écrire une fonction `donne_decimale` qui extrait et renvoie la partie entière du nombre réel (`double`) passé en argument (fonction `floor` de `#include<cmath>`), tout en modifiant ce nombre pour le préparer pour l'extraction de la décimale suivante.
2. Faire afficher par des `cout` les 20 premières de π (constante `M_PI` donnée par `cmath`)¹.

1. Sous Visual Studio, il faut insérer dans le fichier source avant toute chose la ligne `#define _USE_MATH_DEFINES`

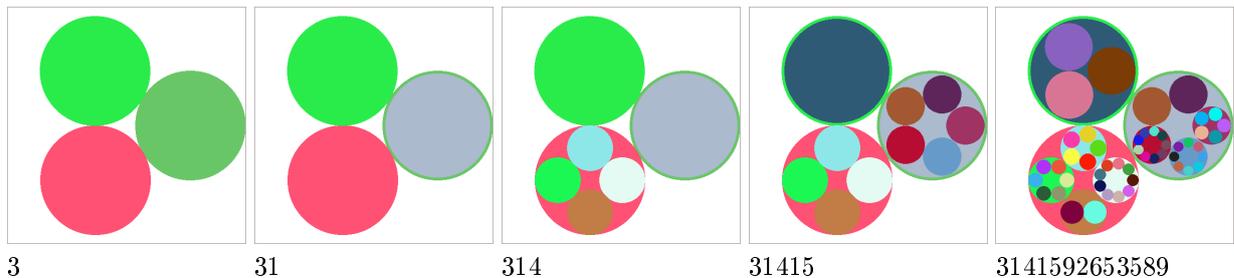


FIGURE 1 – Représentation progressive des décimales de π .

1.1.2 File d'attente

Tout ce qui concerne cette section doit être fait dans des fichiers séparés `file.h` et `file.cpp`.

3. Définir une structure `Cercle` comportant une position (x, y) et un rayon (tous des entiers).
4. Définir une structure `File` définissant une file d'attente. Il s'agit d'un tableau de 300 `Cercles` et d'un indice n indiquant combien d'éléments il y a dans la file. Ces éléments seront stockés des indices 0 à $n - 1$ du tableau.
5. Définir la fonction `push` qui ajoute un élément en queue dans la file.
6. Définir la fonction `pop` qui retire et renvoie l'élément en tête de la file.
7. Protéger ces deux fonctions par des `assert` appropriés pour s'assurer qu'on ne fait pas de bêtise.

1.1.3 Affichage des disques

Le reste du code s'écrit dans le fichier principal.

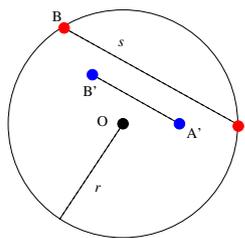
8. Écrire une fonction `affiche_chiffre` qui prend une file et un entier n (entre 0 et 9). On commence par extraire le cercle c en tête de la file (notons (x, y) son centre et r son rayon), et qui désigne la zone d'affichage. Soit s la distance euclidienne entre les points $(r, 0)$ et $(r \cos \frac{2\pi}{n}, r \sin \frac{2\pi}{n})$. Le rayon des cerles à tracer est $r' = \frac{r*s/2}{r+s/2}$. Leur centre est en $(x + (r - r') \cos \frac{2i\pi}{n}, y + (r - r') \sin \frac{2i\pi}{n})$, i étant l'indice du cercle. Si $n = 1$ on prend pour r' 95% de r , et on garde le même centre. Afficher, avec une couleur aléatoire, ces disques (fonction `fillCircle`), et les insérer dans la file d'attente.
9. Écrire une fonction `affiche_nombre`, qui affiche une à une avec `affiche_chiffre` les 30 premières décimales du nombre passé en paramètre, attendant un clic souris de l'utilisateur entre chacune. En même temps, on affichera avec `cout << n << flush;` la décimale courante². On commence par mettre dans la file (sans afficher) un `Cercle` centré au milieu de la fenêtre d'affichage carrée et tangent aux cotés.

1.1.4 Mise en pratique

10. Faire afficher ainsi les décimales de π .
11. Générer un nombre dont les 30 décimales sont tirées au hasard et le faire afficher.

Important : Quand vous avez terminé, créez une archive du projet à votre nom et numéro de groupe en ZIP, RAR, TGZ ou 7z. N'incluez que les fichiers source et le `CMakeLists.txt`, pas les fichiers binaires créés par `Cmake`. Ne partez pas avant que le surveillant ne soit passé vous voir pour copier cette archive sur clé USB.

Annexe : preuve des formules de la question 8



Entre les points $A = (r, 0)$ et $B = (r \cos \frac{2\pi}{n}, r \sin \frac{2\pi}{n})$, on a la distance s . Les triangles AOB et $A'OB'$ sont proportionnels de facteur λ , A' et B' étant les centres des cercles internes. Ces cercles devant être tangents, leur rayon doit être $A'B'/2 = \lambda AB/2 = \lambda s/2$. On a alors $r = OA = OA' + A'A = \lambda r + \lambda s/2$, d'où $\lambda = \frac{r}{r+s/2}$, donc $r' = \frac{r*s/2}{r+s/2}$.

2. le `flush` sert à forcer l'affichage sans revenir à la ligne, contrairement à `endl`