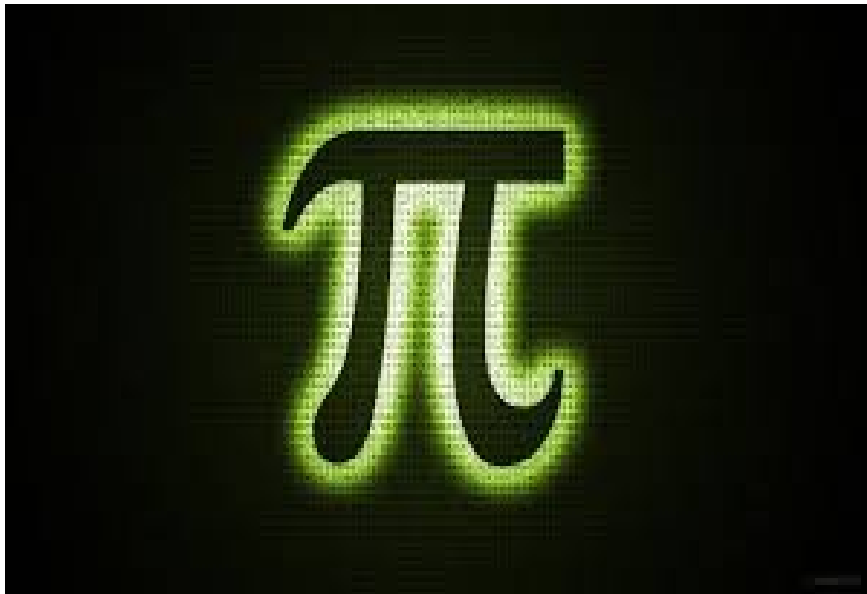


*Machines parallèles*  
*Algorithme BBP  $\pi$*



*Suivi par Boucheta Mehdi*

**Nom** : TACHOUR

**Prénom** : Omar

**Num d'étudiant** : 16706265

## Résumé du projet :

Mon programme consiste à appliquer l'algorithme BBP(ou formule de Bailey-Borwein-Plouffe) pour calculer l'approximation de PI après la virgule tout dépend de k à donner pendant l'exécution.

$$\pi = \sum_{k=0}^{\infty} \frac{1}{16^k} \left( \frac{4}{8k+1} - \frac{2}{8k+4} - \frac{1}{8k+5} - \frac{1}{8k+6} \right)$$

Le programme se compose de trois fichiers « main.cpp » et «bbp.cpp » et « header.hpp ».

1/« **bbp.cpp** » : se compose d'un ensemble de fonctions qui permet la réalisation de l'algorithme BBP :

- **long double algo\_bbp(int debut, int fin) :**
  - C'est la fonction qui permet d'appliquer la formule bbp pour chaque i de la boucle for, confiné entre 0 et k.
  - Elle prend en paramètre deux entiers (debut et fin).
    - début :c'est le début de chaque partie située dans l'intervalle [0 , k] gérée par la fonction « mult\_thread» .
    - fin qui est la fin de chaque partie.
  - Enfin elle affiche le i et le « pi » de chaque i de la boucle.
- **void mult\_thread() :**c'est la fonction qui permet de générer les threads sachant que chaque thread travaille sur une partie de l'intervalle [0 à k], sachant que chaque partie est délimitée entre begin et end.
- **int recorder() :** c'est la fonction principale de programme c'est là ou on fait appel à toutes les fonctions du programme.

2/ « **main.cpp** » : c'est là où la fonction recorder est appelée.

3/ « **header.hpp** » : il contient les library nécessaires pour réaliser le programme, et les entêtes de toutes les fonctions définies dans le code.

4/ « **makefile** » .

5/ « **resultat.txt** » : il contient quelques exemples d'exécution.