

Program main.c:

```
/**
 * @file main.c
 * @author Durand Thomas
 * @brief fonction main permettant d'appeler les differents calcul a travers un menu de selection
 * @version 0.1
 * @date 2019-10-28
 *
 * @copyright Copyright (c) 2019
 *
 */

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <time.h>
#include "PI.h"
#include "sqrt.h"

/**
 * @brief menue d'appel des different calculs
 *
 * @return int
 */
int main(){

    printf("Bonjour que puis-je faire pour vous\n");
    printf("1 pour calculer PI avec MonteCarlo \n");
    printf("2 pour calculer PI avec Madhava \n");
    printf("3 pour calculer PI avec John Wallis \n");
    printf("4 pour calculer rac(2) avec Newton \n");
    printf("5 pour calculer rac(2) avec Halley \n");
    printf("6 pour calculer rac(2) avec Theon \n");

    int rep,n;
    float res;
    scanf("%d",&rep);

    switch (rep){
    case 1:
        printf("combien d'iteration?\n");
```

```

    scanf("%d",&n);
    res = calculePiMonteCarlo(n);
    printf("Le calcule donne: %f\n",res);
    break;
case 2:
    printf("combien d'iteration?\n");
    scanf("%d",&n);
    res = calculePiMadhava(n);
    printf("Le calcule donne: %f\n",res);
    break;
case 3:
    printf("combien d'iteration?\n");
    scanf("%d",&n);
    res = calculePiJohnWallis(n);
    printf("Le calcule donne: %f\n",res);
    break;
case 5:
    printf("combien d'iteration?\n");
    scanf("%d",&n);
    res = methodeNewton(n);
    printf("Le calcule donne: %f\n",res);
    break;
case 6:
    printf("combien d'iteration?\n");
    scanf("%d",&n);
    res = methodeHalley(n);
    printf("Le calcule donne: %f\n",res);
    break;
case 7:
    printf("combien d'iteration?\n");
    scanf("%d",&n);
    res = methodeTheon(n);
    printf("Le calcule donne: %f\n",res);
    break;
default:
    printf("il y a eu une erreur desole \n");
    break;
}

return 0;
}

```

```

/**
 * @file main.c
 * @author Durand Thomas
 * @brief fonction main permettant d'appeler les differents calcul a travers un menu de selection
 * @version 0.1
 * @date 2019-10-28
 *
 * @copyright Copyright (c) 2019
 *
 */

```

```

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <time.h>
#include "PI.h"
#include "sqrt.h"

```

```

/**
 * @brief menue d'appel des different calculs
 *
 * @return int
 */

```

```

int main(){

    printf("Bonjour que puis-je faire pour vous\n");
    printf("1 pour calculer PI avec MonteCarlo \n");
    printf("2 pour calculer PI avec Madhava \n");
    printf("3 pour calculer PI avec John Wallis \n");
    printf("4 pour calculer rac(2) avec Newton \n");
    printf("5 pour calculer rac(2) avec Halley \n");
    printf("6 pour calculer rac(2) avec Theon \n");

    int rep,n;
    float res;
    scanf("%d",&rep);

    switch (rep){
    case 1:
        printf("combien d'iteration?\n");
        scanf("%d",&n);
        res = calculePiMonteCarlo(n);

```

```

    printf("Le calcule donne: %f\n",res);
    break;
case 2:
    printf("combien d'iteration?\n");
    scanf("%d",&n);
    res = calculePiMadhava(n);
    printf("Le calcule donne: %f\n",res);
    break;
case 3:
    printf("combien d'iteration?\n");
    scanf("%d",&n);
    res = calculePiJohnWallis(n);
    printf("Le calcule donne: %f\n",res);
    break;
case 5:
    printf("combien d'iteration?\n");
    scanf("%d",&n);
    res = methodeNewton(n);
    printf("Le calcule donne: %f\n",res);
    break;
case 6:
    printf("combien d'iteration?\n");
    scanf("%d",&n);
    res = methodeHalley(n);
    printf("Le calcule donne: %f\n",res);
    break;
case 7:
    printf("combien d'iteration?\n");
    scanf("%d",&n);
    res = methodeTheon(n);
    printf("Le calcule donne: %f\n",res);
    break;
default:
    printf("il y a eu une erreur desole \n");
    break;
}

return 0;
}

```

Fichier sqrt.c

```

/**
 * @file sqrt.c
 * @author Durand Thomas
 * @brief ensemble des fonctions pour l'aproximation de rac(2)
 * @version 0.1
 * @date 2019-10-14
 *
 * @copyright Copyright (c) 2019
 *
 */
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <time.h>

/**
 * @brief aproxime rac(2) avec la methode de Newton
 *
 * @param n
 * @return float
 */
float methodeNewton(int n){
    float res = (1/2)+(1);
    for (int i = 0; i < n; i++){
        res = (res/2) + (1/res);
    }
    return res;
}

/**
 * @brief aproxime rac(2) avec la methode de Newton
 *
 * @param n
 * @return float
 */
float methodeHalley(int n){
    float res = (1+6)/(3+2);
    for (int i = 0; i < n; i++){
        res = res * (res * res + 6) / ( 3 * res * res + 2 );
    }
    return res;
}

```

```

/**
 * @brief aproxime rac(2) avec la methode de Theon
 *
 * @param n
 * @return float
 */
float methodeTheon(int n){
    double pn = 1 + 2;
    double qn = 1 + 1;
    float temp;
    for (int i = 0; i < n; i++){
        temp = pn;
        pn = pn + 2 * qn;
        qn = temp + qn;
    }

    return (pn / qn);
}

```