```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>


double Sn(int n) {
  int i,j=1;
  double x=3;
  for (i=1; i<=n; i++) { x += j * 4.0/(2*i)/(2*i+1)/(2*i+2); j *= -1; }
  return x;
}

double Pi(double eps) { double x,y;
  int i=1;
  y=Sn(i);
  do { x=y; y=Sn(++i); } while ( fabs(x-y)>eps*(fabs(x)+fabs(y)));
  return y;
}

void ex1() { double e, pi;
  printf("Entrer la precision "); fflush(stdout);
  scanf("%lf",&e);
  printf("Valeur de Pi : %1.16lf\n",Pi(e));
}

void produit(double a, double b, double* pr) { *pr = a * b; }

double somme(double a, double b) {return a+b; }

void sommeprod(double a, double b, double* ps, double* pp) {
  produit(a,b,pp);
  *ps=somme(a,b);
}

void ex2() {
  double x,y,z,t;
  printf("Entrer 2 reels "); fflush(stdout);
  scanf("%lf %lf",&x,&y);
  sommeprod(x,y,&z,&t);
  printf("Valeurs calculees : %lf %lf \n",z,t);
}



int** alloueMatrice(int n, int m) { int i;
  int** p = calloc(n,sizeof(*p));
  if (p==NULL) return NULL;
  *p = calloc(n*m,sizeof(**p));
    if (*p==NULL) {free(p); return NULL;}
  for (i=0; i<n-1;i++)
    p[i+1] = p[i]+m;
  return p;
}

void libereMatrice(int** m) { free(*m); free(m); }

int testeLigne(int** carre, int nl, int nc, int k) { int i;
  int* hist = calloc(nc,sizeof(*hist));
  for(i=0; i<nc; i++)
```

```c
    hist[carre[k][i]]++;
  for(i=0; i<nc; i++)
    if (hist[i]!=1) { free(hist); return 0; }
  free(hist);
  return 1;
}

int testeLignePointeurs(int** carre, int nl, int nc, int k) { int* p;
  int* hist = calloc(nc,sizeof(*hist));
  for(p=carre[k]; p<carre[k]+nc; p++)
    hist[*p]++;
  for(p=hist; p<hist+nc; p++)
    if (*p!=1) { free(hist); return 0; }
  free(hist);
  return 1;
}

int testeColonne(int** carre, int nl, int nc, int k) { int i;
  int* hist = calloc(nl,sizeof(*hist));
  for(i=0; i<nl; i++)
    hist[carre[i][k]]++;
  for(i=0; i<nl; i++)
    if (hist[i]!=1) {  free(hist); return 0; }
  free(hist);
  return 1;
}

int testeColonnePointeurs(int** carre, int nl, int nc, int k) { int* p;
  int* hist = calloc(nl,sizeof(*hist));
  for(p=carre[0]+k; p<=carre[nl-1]+k; p+=nc)
    hist[*p]++;
  for(p=hist; p<hist+nc; p++)
    if (*p!=1) {  free(hist); return 0; }
  free(hist);
  return 1;
}

int estLatin(int** carre, int n) {
  int i;
  for (i=0; i<n; i++)
    if (testeLignePointeurs(carre,n,n,i)==0 || testeColonnePointeurs(carre,n,n
        ,i)==0) {return 0; }
  return 1;
}

int** genereCarre(int n) { int** carre; int i,j;
  carre=alloueMatrice(n,n);
  for(i=0; i<n; i++)
    for(j=0; j<n; j++)
      carre[i][j]=(i+j)%n;
  return carre;
}

void afficheCarre(int** carre, int n) { int i,j;
  for(i=0; i<n; i++)  {
    for(j=0; j<n; j++)
      printf("%3d ", carre[i][j]);
    printf("\n");;
  }
}
```

```c
void sauveCarreTexte(char* nomfichier,int** carre, int n) { int i,j;
  FILE* fp;
  if ( (fp=fopen(nomfichier,"w")) ==NULL) return;
  fprintf(fp,"%d\n",n);
  for(i=0; i<n; i++)  {
    for(j=0; j<n; j++)
      fprintf(fp,"%d ", carre[i][j]);
    fprintf(fp,"\n");;
  }
  fclose(fp);
}

void sauveCarreBinaire(char* nomfichier,int** carre, int n) { int i,j;
  FILE* fp;
  if ( (fp=fopen(nomfichier,"wb")) ==NULL) return;
  fwrite(&n,1,sizeof(n),fp);
  fwrite(*carre,sizeof(**carre),n*n,fp);
  fclose(fp);
}

int** chargeCarre(char* nomfichier, int*  pn) { int i,j;
  FILE* fp;
  int** carre;
  if ( (fp=fopen(nomfichier,"rb")) == NULL) return NULL;
  fread(pn,1,sizeof(*pn),fp);
  if ( (carre=alloueMatrice(*pn,*pn)) == NULL) return NULL;
  fread(*carre,*pn * *pn,sizeof(**carre),fp);
  fclose(fp);
  return carre;
}

void ex3() {
  int n;
  int** c;
  int** d;
  printf("Entrer la dimension du carre latin "); fflush(stdout); scanf("%d",&n
      );
  c=genereCarre(n);
  afficheCarre(c,n);
  if (estLatin(c,n)) printf("C'est un carre latin\n");
  else printf("Ce n'est pas un carre latin\n");
  sauveCarreTexte("carre.txt",c,n);
  sauveCarreBinaire("carre.bin",c,n);
  d=chargeCarre("carre.bin",&n);
  puts("----------");
  afficheCarre(d,n);

  libereMatrice(c);
  libereMatrice(d);
}


main() {
  ex1();
  ex2();
  ex3();
}
```