```
procédure afficheRegle
//procédure pour afficher les règles
//paramètre : grille (entrée) chaine : afficher une chaine de caractères
//résultat : les règles du jeux

procédure afficheGrille
//procédure pour afficher la grille
//paramètre : (entrée) chaine : afficher une chaine de caractères
//résultat : La grille du sudoku


fonction premiereSaise
//fonction pour effectuer la première saisie, retourne la case séléctionnée
//paramètre : ligne (entrée) chaine : séléctionner une ligne
//paramètre : colonne (entrée) entier : séléctionner une colonne
//résultat : séléction d'une case de la grille

procédure erreurPremiereSaisie
//procédure pour les erreurs de la première saisie
//paramètre : ligne (entrée) reel : séléctionner une ligne
//paramètre : colonne (entrée) chaine : séléctionner une colonne
//paramètre : colonne (entrée) reel : séléctionner une colonne
//résultat : message d'erreur si l'utilisateur saisis les aruments dans le
mauvais ordre,
//si les arguments sont innexistants ou si l'argument pour la colone est un
nombre à virgule

fonction deuxiemeSaisie
//fonction pour effetuer la deuxième saisie, retourne la valeur saisie
//paramètre : valeur (entrée) entier : saisir une valeur
//paramètre : changerCase (entrée) entier : changer de case
//paramètre : abandonner (entrée) chaine : abandonner la partie
//paramètre : remplacer (entrée) chaine : remplacer une valeur de la grille
//résultat : valeur entrée dans la grille OU choix d'une autre case OU abandon
de la partie
//OU une valeur est supprimée

procédure erreurDeuxiemeSaisie
//procédure pour les erreurs de la deuxième saisie
//paramètre : mauvaiseValeur (entrée) entier : saisir une valeur impossisble à
poser
//paramètre : valeurInnexistante (entrée) entier : saisir une valeur qui
n'existe pas
//paramètre : caractèreInnexistant (entrée) chaine : saisir un caractère qui
n'existe pas
//paramètre : valeurVirgule (entrée) reel : saisir une valeur à virgule
//paramètre : remplaceInitial (entrée) chaine : remplacer une valeur initiale
//paramètre : caseRemplie (entrée) entier : saisir une valeur sur une case
remplie
//résultat : message d'erreur si l'utilisateur saisis une valeur déja présente
OU si la valeur
//saisie est innexistante OU si un caractère saisis est innexistant OU si une
valeur à virgule
//est saisie OU si une valeur initiale veut être remplacée OU si une case est
déja remplie et que
//le joueur souhaite la remplir sans en enlever le contenu

fonction troisiemeSaisie
//fonction pour effectuer la troisieme saisie, retourne la valeur saisie
//paramètre : valeur (entrée) entier : saisir une valeur
//paramètre : changerCase (entrée) entier : changer de case
//paramètre : abandonner (entrée) chaine : abandonner la partie
//résultat : valeur entrée dans la grille OU choix d'une autre case OU abandon
```

```
de la partie

procédure erreurTroisiemeSaisie
//procédure pour les erreurs de la deuxième saisie
//paramètre : mauvaiseValeur (entrée) entier : saisir une valeur impossisble à
poser
//paramètre : valeurInnexistante (entrée) entier : saisir une valeur qui
n'existe pas
//paramètre : caractèreInnexistant (entrée) chaine : saisir un caractère qui
n'existe pas
//paramètre : valeurVirgule (entrée) reel : saisir une valeur à virgule
//paramètre : caseRemplie (entrée) entier : saisir une valeur sur une case
remplie
//résultat : message d'erreur si l'utilisateur saisis une valeur déja présente
OU si la valeur
//saisie est innexistante OU si un caractère saisis est innexistant OU si une
valeur à virgule
//est saisie OU si une case est déja remplie et que
//le joueur souhaite la remplir sans en enlever le contenu

procédure finJeux
//procédure pour la fin du jeux
//paramètre : abandon (entrée) chaine : abandon, message de fin
//paramètre : gagne (entrée) entier : grille remplie, message de fin
//résultat : jeux temriné, message de fin


//programme principal
programme SUDOKU c'est
début
    constante changeCase = 0;
    constante abandonner = Q;
    constante remplacer = R;

    regle : chaine;
    grille : chaine;
    ligneSaisie : chaine;
    colonneSaisie : entier;
    erreurLigne : reel;
    erreurcolonneVirgule : reel
    erreurColonne : chaine;
    valeurSaisieDeux : entier;
    mauvaiseValeur : entier;
    valeurInnexistante : reel;
    caractèreInnexistant : chaine;
    valeurVirgule : reel;
    remplaceInitial : chaine;
    caseRemplie : entier;
    valeurSaisieTrois : entier;

    regle := afficheRegle;
    affiche(regle);
    grille := afficheGrille;
    affiche(grille);
    resultatSaisieUne := premiereSaisie(ligneSaisie, colonneSaisie);
    affiche(resultatSaisieUne);
    si(resultatSaisieUne = vrai) alors
        resultatSaisieDeux := deuxiemeSaisie(valeurSaisieDeux; changeCase;
abandonner; remplacer);
        affiche(resultatSaisieDeux);
        si(resultatSaisieDeux = vrai) alors
        resultatSaisieTrois := TroisiemeSaisie(valeurSaisieTrois; changeCase;
abandonner);
        affiche(resultatSaisieTrois);
```

```
            si(resultatSaisieTrois = vrai) alors
                finDuJeux := finJeux(abandonner; gagne);
                affiche(finDuJeux);
                si(finDuJeux = faux) alors
                    retourne resultatSaisieUne;
                finsi
            sinon
                erreurTroisieme := (mauvaiseValeur; valeurInnexistante;
caractèreInnexistant;  valeurVirgule; caseRemplie);
                affiche(erreutTroisieme);
            finsi
        sinon
            erreurDeuxieme := erreurDeuxiemeSaisie(mauvaiseValeur;
valeurInnexistante; caractèreInnexistant;  valeurVirgule; remplaceInitial;
caseRemplie);
            affiche(erreurDeuxieme);
        finsi
    sinon
        erreurPremiere := erreurPremiereSaisie(erreurLigne;erreurColonne;
erreurColonneVirgule);
        affiche(erreurPremiere);
    finsi

fin
```