

programme yams c'est

```
constante entier MAXDES := 5;
constante entier MAXSCORE := 50;
constante entier BONUS := 35;
constante entier FULLHOUSE:= 25;
constante entier PETITESUITE := 30;
constante entier GRANDESUITE := 40;
constante entier YAMS := 50;
```

```
procédure demandeNomJoueur(sortF nomJ1: chaine(20), sortF nomJ2: chaine(20));
procédure afficheFeuilleDeMarque(entF nomJ1: chaine(20), entF nomJ2: chaine(20),entF
tableauDeScoreJ1: tableau [] de entier,entF tableauDeScoreJ2 : tableau [] de entier);
procédure demandeLancerDés(sortF réponseLancer: chaine(20));
procédure lancerDés(entF/sortF tableauDeDes: tableau[] de entier,entF nbreRelanceDés : tableau[]
de entier );
procédure affichageDés(entF tableauDeDés);
procédure choixDés(sortF nbreRelanceDés);
procédure choixRelance(entE/sortE réponseRelancer : entier);
procédure affecterScore(entF tableauDeDés, entF/sortF scoreTotalJ : entier,entF/sortF
tableauDeScoreJ);
procédure afficheVainqueur(entF scoreTotalJ1 : entier, entF scoreTotalJ2 : entier);
```

début

```
i,k,scoreTotalJ1,scoreTotalJ2,nbreLancer: entier;
tableauDeDés : tableau[1..MAXDES] de entier;
tableauDeScoreJ1 : tableau[1..MAXSCORE] de entier;
tableauDeScoreJ2 : tableau[1..MAXSCORE] de entier;
nomJ1 : chaine(20);
nomJ2 : chaine(20);
réponseLancer : chaine(10);
réponseRelancer : chaine(10);
```

```
demandeNomJoueur(sortE nomJ1,sortE nomJ2);
scoreTotalJ1 := 0;
scoreTotalJ2 := 0;
```

```
pour i de 1 à 13 pas 1 faire
  k := 1;
  afficheFeuilleDeMarque(entE nomJ1, entE nomJ2,entE tableauDeScoreJ1,entE
tableauDeScoreJ2);
  pour k de 1 à 2 pas 1 faire
    nbreLancer = 2;
    si (k == 1) alors
      demandeLancerDés(sortE réponseLancer);
      si (reponse == "oui") alors
        lancerDés(entE/sortE tableauDeDés,entE nbreRelanceDés );
```

```

    affichageDés(entE tableauDeDés);
    choixRelance(sortE réponseRelancer);
    tant que (réponseRelancer == "oui" ET nbreLancer != 0) faire
        choixDés(sortE nbreRelanceDés);
        lancerDes(entE/sortE tableaudeDés,entE nbreRelanceDés );
        affichageDés(entE tableauDeDés);
        choixRelance(sortE réponseRelancer);
        nbreLancer = nbreLancer - 1;
    fin faire
    affecteScore(entE tableauDeDés,entE/sortE scoreTotalJ1,entE/sortE
tableauDeScoreJ1);
    sinon
        écrireEcran(nomJ2,"gagne par forfait");
        i:= 13;
    fin si
sinon
    demandeLancerDés(sortE réponseLancer)
    si (reponse == "oui") alors
        lancerDés(entE/sortE tableauDeDés,entE nbreRelanceDés );
        affichageDés(entE tableauDeDés);
        choixRelance(sortE réponseRelancer);
        tant que (réponseRelancer == "oui" ET nbreLancer != 0) faire
            choixDés(sortE nbreRelanceDés);
            lancerDes(entE/sortE tableauDeDés,entE nbreRelanceDés );
            affichageDés(entE tableauDeDés);
            choixRelance(sortE réponseRelancer);
            nbreLancer = nbreLancer - 1;
        fin faire
        affecteScore(entE tableauDeDés,entE/sortE scoreTotalJ2,entE/sortE tableauDeScoreJ2);
    sinon
        écrireEcran(nomJ1,"gagne par forfait");
        i := 13;
    fin si
    fin si
    fin faire
    fin faire
    afficheVainqueur(entE scoreTotalJ1,entE scoreTotalJ2);
fin

```

//Liste des procédures et fonctions

procédure demandeNomJoueur

paramètre

--sortie--

nomJ1: Stock le nom du premier joueur

nomJ1: Stock le nom du deuxième joueur

Utilité:

Permet de demander correctement un à un le nom des joueurs(nom différent, puis respectant la limite de caractères autorisé)

procédure afficheFeuilleDeMarque

paramètre

--entrée--

nomJ1 : Stock le nom du premier joueur à afficher

nomJ2 : Stock le nom du deuxième joueur à afficher

tableauDeScoreJ1 : Stock les scores du joueur 1 pour chaque combinaison

tableauDeScoreJ2 : Stock les scores du joueur 2 pour chaque combinaison

Utilité:

Affiche le nom des joueurs, puis leurs scores respectif sur la feuille de marque selon l'avancée de la partie

procédure demandeLancerDés

paramètre

--sortie--

réponseLancer : Stock la réponse du joueur

Utilité:

Affecte correctement la réponse du joueur pour savoir si il lance les dés ou abandonne(oui/non)

procédure lancerDés

paramètre

--entrée/sortie--

tableauDeDes : Stock le résultat de chaque dés du lancer

Utilité :

Afficher le résultat des 5 dés lors d'un lancé

procédure affichageDés

paramètre

--entrée--

tableauDeDés : Envoie le tableau du résultat d'un lancé de dés (5 valeurs)

Utilité :

Affiche à l'écran le résultat d'un lancé de dés

procédure choixDés

paramètre

--sortie--

nbreRelanceDés : tableau qui précise les dés que l'on veut relancer

Utilité :

Permet de ne relancer que les dés souhaiter

procédure choixRelance

paramètre

--entrée/sortie--

RéponseRelance : Stock la réponse d'un joueur

Utilité:

Effectue correctement le choix du joueur(relance(oui),préserve(non))

procédure affecteScore

paramètre

--entrée--

tableauDeDés : Transmet la valeur des 5 dés finaux du tour d'un joueur

--entrée/sortie--

scoreTotalJ : Mets à jour le score total du joueur préciser

tableauDeScoreJ : Affecte le nouveau score pour une combinaison à la suite des précédentes, pour un joueur préciser

Utilité :

Permet de choisir une combinaison et de mettre à jour le score total et le tableau des scores du joueur préciser

procédure afficheVainqueur

paramètre

--entrée--

scoreTotalJ1 : Transmet les points totaux du joueur 1

scoreTotalJ2 : Transmet les points totaux du joueur 2

Utilité :

Compare les 2 scores et affiche qui est le vainqueur