Le problème du sac à dos suppose que l’on ait à choisir, parmi un certain nombre d’articles, ceux que l’on se propose de porter sur le dos. Chacun possède un certain poids et une certaine utilité. Or, il faut effectuer un choix de façon à ne pas dépasser un certain poids limite ( le poids maximal autorisé). Le problème consiste dans le même temps à maximiser l’utilité globale des objets emportés. On peut simplifier le problème de deux manières. Premièrement, on s’obligera à atteindre exactement le maximum du poids autorisé et, deuxièmement, on supposera que l’utilité réelle de chaque objet est proportionnelle à son poids. Ceci signifie que l’on doit si possible de trouver un sous-ensemble constitué d’articles dont le poids cumulé est égale à la limite permise. Il est évident qu’on peut rapprocher ce problème de nombreux autres, comme par exemple la façon de choisir dans le porte-monnaie un certain jeu de pièces et de billets pour s’acquitter d’une somme donnée. En général, à partir d’une somme s et de n entiers positifs.

d0, d1,…,dn-1

le problème consiste à trouver un sous-ensemble de ces entiers ( chacun figurant une fois au maximum) dont le total cumulé vaut s.

pour résoudre ce problème, écrire une fonction récursive **sommexac**, appelé e dans le programme principal de la façon suivante :

if( sommexac(s,0)) puts(« \nProblème résolu”);

else puts(\nProblème insoluble”);

En général, l’appel **sommexac(t,i)** a pour but de vérifier si la cible t peut être constituée en additionnant certains entiers choisis dans la suite :

di, di+1,…, dn-1

lorsque cela est possible, les entiers choisis doivent être affichés et la valeur du code-retour est 1 ; dans le cas contraire, on doit renvoyer la valeur 0.

Si t=0, le problème est résolu en utilisant un sous ensemble vide ; il n’existe pas de solution lorsque t<0 ou i=n.

Lorsque t est positive et i inférieur à n, **sommexac**, ( munie des arguments t et i) détermine si la cible t- di peut être formée à partir de la somme de quelques entiers choisis dans la suite : di+1,…, dn-1 ;

On utilise l’appel récursif : **sommexac(t – d[i],i+1)**