

Contraintes

Les solutions au DS sont validées via la plateforme domjudge. Chaque élève a son propre login et son propre mot de passe. Vous pouvez envoyer plusieurs fois une solution pour un même problème sans aucune pénalité.

Vous avez le droit aux supports des cours, TD, TP et aussi à des ressources Internet, mais vous n'avez pas le droit de communiquer avec d'autres personnes. La DSI sauvegarde tout le trafic TCP/IP pendant le DS. Vous pouvez utiliser seulement le protocole http pour accéder aux différents sites Web. Une connexion à un outil de messagerie, réseau social etc invalide automatiquement votre participation au DS avec les conséquences prévues par le règlement des études.

Vous pouvez utiliser le langage C ou le C++, mais sans utiliser la STL.

Problème 4 : Sac vert

Nous disposons d'un sac de capacité C (nombre entier positif), et nous avons également à notre disposition un ensemble d'objets, coloriés en rouge ou en vert. Chaque objet a un poids (en kilogrammes) sous forme d'un nombre entier positif. Nous souhaitons remplir entièrement le sac avec des objets rouges et verts. Quel est le nombre minimal d'objets verts nécessaires pour obtenir un remplissage valide du sac ? Observation : le remplissage peut utiliser également des objets rouges.

Format en entrée

L'entrée standard sera composée d'une série de nombres séparés par des retours à la ligne :

- un nombre entier positif ou nul indiquant la capacité du sac (noté C par la suite, $C < 1000$)
- un nombre entier positif ou nul indiquant le nombre d'objets (noté n par la suite, $n < 100$)
- n lignes comprenant deux entiers séparés par espace, le premier représente la couleur de l'objet (1 pour vert et 2 pour rouge) et le deuxième le poids de l'objet (nombre entier positif).

Le format en entrée est respecté, vous n'avez pas à faire de test pour le vérifier.

Format en sortie

La sortie standard affichera le nombre minimal d'objets verts utilisés pour remplir le sac de capacité C , sur une seule ligne. Observation : ce numéro peut être 0 ! Cette ligne sera finalisée par un retour à la ligne de type `"\r\n"`. Si le sac ne peut pas être rempli la solution affichée sera -1 suivi de la fin de ligne.

Exemple 1

Entrée :

```
100
4
1 50
1 60
2 70
2 60
```

Sortie (aucune solution) :
-1

Exemple 2

Entrée :

100

6

1 10

1 92

2 30

1 60

2 20

2 30

Sortie (solution correspondant à 10+60+30, les deux premiers objets étant verts) :
2

Entrées-sorties

Pour lire un entier sur l'entrée standard, vous pouvez utiliser le code suivant :

```
#define _CRT_SECURE_NO_WARNINGS //pour Visual Studio

#include <stdio.h>
int n;
fscanf(stdin, "%d", &n);
```

Pour lire une ligne avec deux entiers sur l'entrée standard, vous pouvez utiliser le code suivant :

```
#include <stdio.h>
int n1, n2;
fscanf(stdin, "%d %d", &n1, &n2);
```

Pour afficher un entier n suivi d'un fin de ligne :

```
#include <stdio.h>
int n;
...
printf("%d\n", n);
```

Observation

Ne pas oublier la ligne :
return 0;
dans votre fonction main !