

Tous les programmes doivent être réalisés sous forme d'applications console.
Les formats des données ainsi que des résultats sont à respecter.

PROBLÈME 1

MESSAGE

50 POINTS

Description et tâche

Max passe un message sous forme de courriel à tous ses amis. Dans son courriel il demande à ses amis destinataires de transférer (« forwarder ») ce même message, chacun à ses propres amis, et qui devront eux aussi répéter cette même opération et ainsi de suite.



Le but du programme est de déterminer un ensemble regroupant le nombre le plus élevé possible de personnes recevant le message mais qui ne sont pas directement amies entre elles. (Attention : il peut exister plusieurs ensembles répondant à ces conditions, mais il est demandé d'en indiquer un seul.)

Entrée et sortie du programme

Entrée

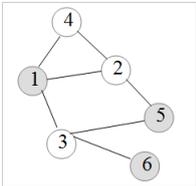
La première ligne contient deux entiers positifs qui sont dans l'ordre n et m . $1 \leq n \leq 50$ représente le nombre de personnes au total et $1 \leq m \leq 1000$ le nombre de relations entre deux personnes (relation « être ami avec »).

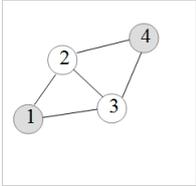
Les m lignes suivantes contiennent chacune deux entiers i et j représentant la relation « i et j sont amis ») avec $i \neq j$ et $1 \leq i < j \leq n$.

Sortie

Le programme affiche deux lignes. La première ligne indique le nombre maximal de personnes ayant reçu le mail mais sans être directement amies entre elles. La deuxième ligne affiche un tel ensemble de personnes.

Exemples d'exécution

Entrée	Sortie
<pre>6 7 1 2 1 3 1 4 2 4 2 5 3 5 3 6</pre> 	<pre>3 1 5 6 (La seule autre solution est ici : 4 5 6)</pre>

Entrée		Sortie
4 5 1 2 1 3 2 3 2 4 3 4		2 1 4

Remettez le programme sous le nom MESSAGE.xxx, avec xxx=PAS ou C(PP) ou JAVA.

PROBLÈME 2

CHAMPIGNONS

50 POINTS

Description

Anne et Bob sont allés à la recherche de champignons dans la forêt. Heureusement pour eux, ils se retrouvent maintenant devant n champignons de taille et qualité variées. Comme ils sont à deux à les ramasser, chacun a emporté un sac à dos. Celui d'Anne a une capacité à emporter c_1 grammes de champignons et celui de Bob c_2 grammes.



Les gourmands ne s'intéressent pas à la quantité de champignons qu'ils emmènent chez eux, mais uniquement à la qualité. Ils sont capables de déterminer pour chaque champignon i avec $1 \leq i \leq n$ sa qualité q_i ainsi que son poids p_i que nous mesurons en grammes. La qualité ainsi que le poids d'un champignon sont toujours des nombres entiers. Ainsi ils essaient de remporter des champignons de telle manière que la somme des qualités de champignons emportés est maximisée sous la contrainte que les champignons soient répartis sur les deux sacs à dos de manière à ce que la somme des tailles des champignons dans un sac à dos soit inférieure ou égale à la capacité de ce sac à dos.

Or, Anne et Bob n'ont pas été très attentifs en classe d'algorithmique et nécessitent donc votre aide à trouver la solution optimale à leur programme.

Tâche

Écrivez un programme qui lit sur la première ligne le nombre de champignons n ainsi que les deux capacités c_1 et c_2 .

Puis suivent n lignes avec sur la ligne $i + 1$ les caractéristiques du champignon i , p_i et q_i , dans cet ordre.

Le programme doit écrire un seul nombre sur la console, précisément le maximum des sommes des qualités des champignons qu'ils peuvent emporter avec leurs sacs à dos.

Attention: Cette tâche est composée de trois sous-tâches, dont les deux premières sont plus faciles. Le problème comme discuté ci-dessus est la sous-tâche 3. Voici la distribution des points ainsi que les contraintes sur les tâches respectives.

Sous-tâche	Contraintes	Points
1	$1 \leq n \leq 15$ $1 \leq c_1 \leq 400$ $c_2 = 0$	5
2	$1 \leq n \leq 400$ $1 \leq c_1 \leq 400$ $c_2 = 0$	15
3	$1 \leq n \leq 400$ $1 \leq c_1 \leq 400$ $1 \leq c_2 \leq 400$	30

Entrées et sorties modèle

Entrée	Sortie
4 10 10 7 6 5 3 5 9 3 2	20

Entrée	Sortie
4 10 0 5 4 4 3 4 3 2 2	8

Explication

Dans le premier exemple Anne peut emporter le premier et le dernier champignon, Bob le deuxième et le troisième. Ils utilisent entièrement les capacités de leurs sacs à dos. Ensemble les champignons auront une qualité cumulée de $6 + 3 + 9 + 2 = 20$.

Dans le deuxième exemple le sac à dos de Bob ne peut contenir aucun champignon à cause de sa capacité nulle. Il est donc à Anne d'emporter les trois derniers champignons pour obtenir la qualité cumulée maximale de $3 + 3 + 2 = 8$.

Remettez le programme sous le nom CHAMPIGNONS.xxx, avec xxx=PAS ou C(PP) ou JAVA.