

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 numéro 1 : la température négative moyenne

Vous disposez de l'historique des températures à Lyon prises chaque heure. Vous devez calculer la température moyenne pour toutes les valeurs négatives de température (≤ 0). Attention : 0 est considéré comme une valeur négative dans ce problème !

Description de l'entrée (IN) : la première ligne va contenir le nombre de valeurs en entrée et les lignes suivantes correspondent aux valeurs de température, une par ligne (valeurs réelles).
Description de la sortie (OUT) : un nombre représentant la moyenne des températures négatives.
Si dans les données d'entrée il n'y a pas de température négative, affichez le caractère '-'.
Observation : chaque ligne affichée sera terminée par deux caractères (CR et LF), correspondant à la chaîne `{"\r\n"}`.

Exemple 1

IN:

3
10
-5
1

OUT:

-5

Exemple 2

IN:

2
10
1

OUT:

-

Exemple 3

IN:

7
0.2
-0.1
-1.3
-0.2

0.3
0
1

OUT:
-0.4

Exemple de code pour les entrées/sorties

```
#include <iostream>
using namespace std;

int main()
{
    int nMeasures = 0;
    cin >> nMeasures;

    for (int nPos = 0; nPos < nMeasures; nPos++)
    {
        float fValue = 0;
        cin >> fValue;
    }

    float fAverage = 0;
    //votre code ...
    // ...

    cout << fAverage << "\r\n";
    return 0;
}
```