

OBI 2013 – Nível Júnior: Fase 1

Saldo do Vovô

Prof. Edson Alves

Faculdade UnB Gama

Vovô João tem uma banca de jornais; ele tem muitos clientes, e diariamente recebe muito dinheiro, mas também faz muitos pagamentos para manter o seu estoque de jornais e revistas. Todo dia ele vai ao banco realizar um depósito ou uma retirada de dinheiro. Em alguns dias, o saldo de sua conta no banco fica negativo, mas Vovô João tem um acordo com o banco que garante que ele somente é cobrado se o saldo for menor do que um valor pré-estabelecido.

Dada a movimentação diária da conta do banco do Vovô João, você deve escrever um programa que calcule o menor saldo da conta, no período dado.

Entrada

A primeira linha da entrada contém dois números inteiros N e S que indicam respectivamente o número de dias do período de interesse e o saldo da conta no início do período. Cada uma das N linhas seguintes contém um número inteiro indicando a movimentação de um dia (valor positivo no caso de depósito, valor negativo no caso de retirada). A movimentação é dada para um período de N dias consecutivos: a primeira das N linhas corresponde ao primeiro dia do período de interesse, a segunda linha corresponde ao segundo dia, e assim por diante.

Saída

Seu programa deve imprimir uma única linha, contendo um único número inteiro, o menor valor de saldo da conta no período dado.

Restrições

- ▶ $1 \leq N \leq 30$
- ▶ $-10^3 \leq S \leq 10^3$
- ▶ $-10^3 \leq \text{cada movimentação} \leq 10^3$

Exemplo de entrada e saída

Exemplo de entrada e saída

6 -200

Exemplo de entrada e saída

6 -200
↑
de dias

Exemplo de entrada e saída

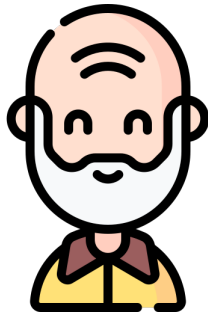
6 -200



saldo inicial

Exemplo de entrada e saída

6 -200



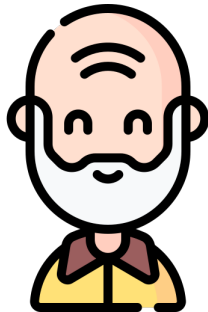
-200

-200

Exemplo de entrada e saída

6 -200

-100



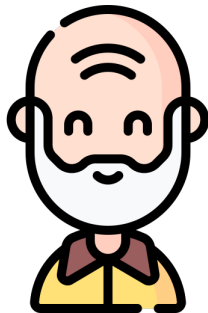
-200

-200

Exemplo de entrada e saída

6 -200

-100 ← retirada no dia 1



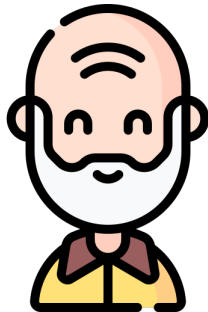
-200

-200

Exemplo de entrada e saída

6 -200

-100



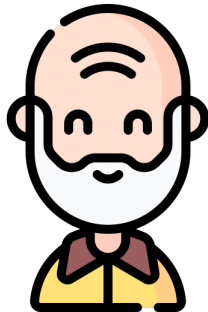
-300

-200

Exemplo de entrada e saída

6 -200

-100



-300

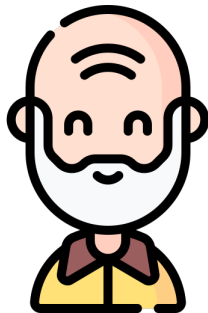
-300

Exemplo de entrada e saída

6 -200

-100

1000



-300

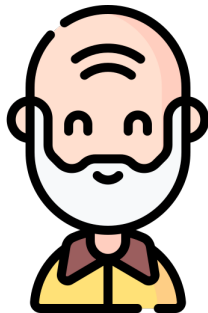
-300

Exemplo de entrada e saída

6 -200

-100

1000 ← depósito no dia 2



-300

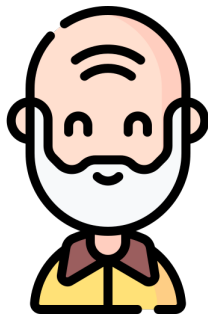
-300

Exemplo de entrada e saída

6 -200

-100

1000



700

-300

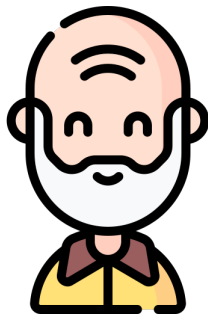
Exemplo de entrada e saída

6 -200

-100

1000

-2000



700

-300

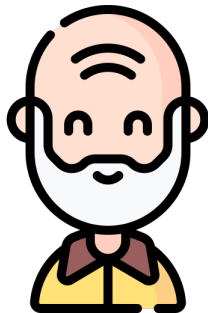
Exemplo de entrada e saída

6 -200

-100

1000

-2000 ← retirada no dia 3



700

-300

Exemplo de entrada e saída

6 -200

-100

1000

-2000



-1300

-300

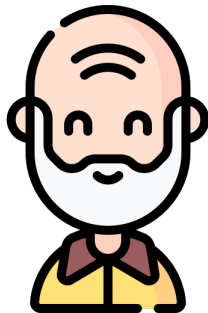
Exemplo de entrada e saída

6 -200

-100

1000

-2000



-1300

-1300

Exemplo de entrada e saída

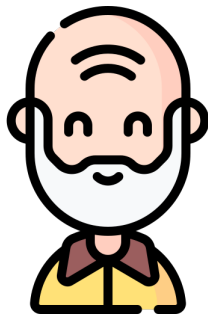
6 -200

-100

1000

-2000

100



-1300

-1300

Exemplo de entrada e saída

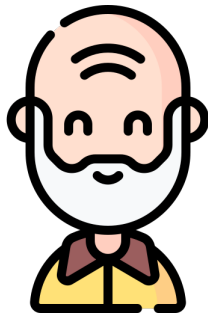
6 -200

-100

1000

-2000

100 ← depósito no dia 4



-1300

-1300

Exemplo de entrada e saída

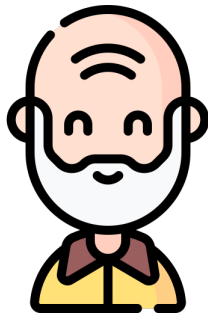
6 -200

-100

1000

-2000

100



-1200

-1300

Exemplo de entrada e saída

6 -200

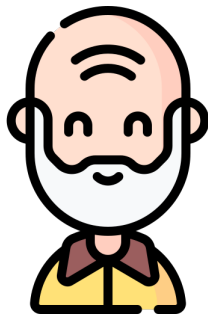
-100

1000

-2000

100

-50



-1200

-1300

Exemplo de entrada e saída

6 -200

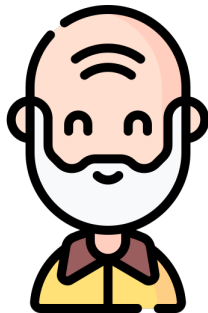
-100

1000

-2000

100

-50 ← retirada no dia 5



-1200

-1300

Exemplo de entrada e saída

6 -200

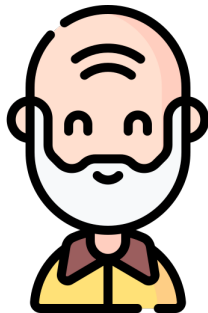
-100

1000

-2000

100

-50



-1250

-1300

Exemplo de entrada e saída

6 -200

-100

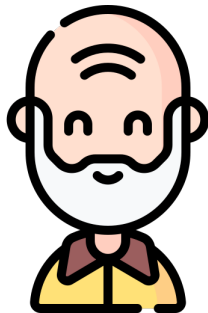
1000

-2000

100

-50

2000



-1250

-1300

Exemplo de entrada e saída

6 -200

-100

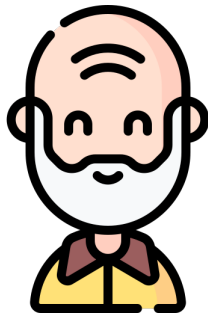
1000

-2000

100

-50

2000 ← depósito no dia 6



-1250

-1300

Exemplo de entrada e saída

6 -200

-100

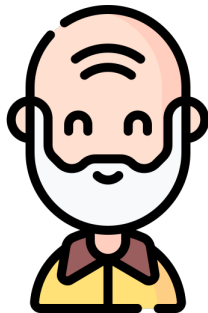
1000

-2000

100

-50

2000



800

-1300

Solução

Solução

- ★ O problema consiste em simular as ações do vovô a cada dia

Solução

- ★ O problema consiste em simular as ações do vovô a cada dia
- ★ Devem ser mantidas duas variáveis: o saldo atual e o menor saldo da série histórica

Solução

- ★ O problema consiste em simular as ações do vovô a cada dia
- ★ Devem ser mantidas duas variáveis: o saldo atual e o menor saldo da série histórica
- ★ É necessário um laço de se repita N vezes para ler a entrada

Solução

- ★ O problema consiste em simular as ações do vovô a cada dia
- ★ Devem ser mantidas duas variáveis: o saldo atual e o menor saldo da série histórica
- ★ É necessário um laço de se repita N vezes para ler a entrada
- ★ A cada inteiro lido, o saldo deve ser atualizado somando-se tal número

Solução

- ★ O problema consiste em simular as ações do vovô a cada dia
- ★ Devem ser mantidas duas variáveis: o saldo atual e o menor saldo da série histórica
- ★ É necessário um laço de se repita N vezes para ler a entrada
- ★ A cada inteiro lido, o saldo deve ser atualizado somando-se tal número
- ★ Se o saldo atual é o menor da série, deve-se atualizar o registro

Solução

- ★ O problema consiste em simular as ações do vovô a cada dia
- ★ Devem ser mantidas duas variáveis: o saldo atual e o menor saldo da série histórica
- ★ É necessário um laço de se repita N vezes para ler a entrada
- ★ A cada inteiro lido, o saldo deve ser atualizado somando-se tal número
- ★ Se o saldo atual é o menor da série, deve-se atualizar o registro
- ★ Esta atualização pode ser feita por meio de um **if** ou da função `min()`

Solução $O(N)$ em C++

```
#include <iostream>
```

```
using namespace std;
```

```
int main()  
{
```

```
    return 0;
```

```
}
```

Solução $O(N)$ em C++

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int N, S;
```

```
        return 0;
```

```
}
```

Solução $O(N)$ em C++

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int N, S;
```

```
    cin >> N >> S;
```

```
        return 0;
```

```
}
```

Solução $O(N)$ em C++

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int N, S;
```

```
    cin >> N >> S;
```

```
    int ans = S;
```

```
}
```

```
    return 0;
```


Solução $O(N)$ em C++

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
}
```

```
    int N, S;
```

```
    cin >> N >> S;
```

```
    int ans = S;
```

```
    return 0;
```

```
}
```

```
    for (int i = 0; i < N; ++i)
```

```
    {
```

Solução $O(N)$ em C++

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
}
```

```
    int N, S;
```

```
    cin >> N >> S;
```

```
    int ans = S;
```

```
    return 0;
```

```
}
```

```
    while (N--)
```

```
{
```

Solução $O(N)$ em C++

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
}
```

```
    int N, S;
```

```
    cin >> N >> S;
```

```
    int ans = S;
```

```
    return 0;
```

```
}
```

```
    while (N--)
```

```
    {
```

```
        int x;
```

Solução $O(N)$ em C++

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
}
```

```
    int N, S;
```

```
    cin >> N >> S;
```

```
    int ans = S;
```

```
    return 0;
```

```
}
```

```
while (N--)
```

```
{
```

```
    int x;
```

```
    cin >> x;
```

Solução $O(N)$ em C++

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int N, S;
```

```
    cin >> N >> S;
```

```
    int ans = S;
```

```
    while (N--)
```

```
    {
```

```
        int x;
```

```
        cin >> x;
```

```
        S += x;
```

```
    }
```

```
    return 0;
```

```
}
```

Solução $O(N)$ em C++

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int N, S;
```

```
    cin >> N >> S;
```

```
    int ans = S;
```

```
    while (N--)
```

```
    {
```

```
        int x;
```

```
        cin >> x;
```

```
        S += x;
```

```
        if (S < ans)
```

```
            ans = S;
```

```
    }
```

```
    return 0;
```

```
}
```

Solução $O(N)$ em C++

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int N, S;
```

```
    cin >> N >> S;
```

```
    int ans = S;
```

```
    while (N--)
```

```
    {
```

```
        int x;
```

```
        cin >> x;
```

```
        S += x;
```

```
        ans = min(ans, S);
```

```
    }
```

```
    return 0;
```

```
}
```

Solução $O(N)$ em C++

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int N, S;
```

```
    cin >> N >> S;
```

```
    int ans = S;
```

```
    while (N--)
```

```
    {
```

```
        int x;
```

```
        cin >> x;
```

```
        S += x;
```

```
        ans = min(ans, S);
```

```
    }
```

```
    cout << ans << "\n";
```

```
    return 0;
```

```
}
```