

# **OBI 2012 – Nível 1: Fase 1**

*Consecutivos*

Prof. Edson Alves

*Faculdade UnB Gama*

Num sorteio que distribui prêmios, um participante inicialmente sorteia um inteiro  $N$  e depois  $N$  valores. O número de pontos do participante é o tamanho da maior sequência de valores consecutivos iguais. Por exemplo, suponhamos que um participante sorteia  $N = 11$  e, nesta ordem, os valores

30, 30, 30, 30, 40, 40, 40, 40, 40, 30, 30

Então, o participante ganha 5 pontos, correspondentes aos 5 valores 40 consecutivos. Note que o participante sorteu 6 valores iguais a 30, mas nem todos são consecutivos.

Sua tarefa é ajudar a organização do evento, escrevendo um programa que determina o número de pontos de um participante.

## Entrada

A primeira linha da entrada contém um inteiro  $N$ , o número de valores sorteados. A segunda linha contém  $N$  valores,  $V_1, V_2, \dots, V_N$ , na ordem de sorteio, separados por um espaço em branco.

## Saída

Seu programa deve imprimir apenas uma linha, contendo apenas um inteiro, indicando o número de pontos do participante.

## Restrições

- ▶  $1 \leq N \leq 10^4$
- ▶  $-2^{31} \leq V_i \leq 2^{31} - 1$ , para  $i = 1, 2, \dots, N$

## **Exemplo de entrada e saída**

## Exemplo de entrada e saída

11

## Exemplo de entrada e saída

11



*# de valores sorteados*

## Exemplo de entrada e saída

11

30 30 30 40 40 40 40 40 30 30 30

## Exemplo de entrada e saída

11

30 30 30 40 40 40 40 40 30 30 30



*Números sorteados*



## Exemplo de entrada e saída

11

30 30 30 40 40 40 40 40 30 30 30



## Exemplo de entrada e saída

11

30 30 30 40 40 40 40 40 30 30 30



## Exemplo de entrada e saída

11

30 30 30 40 40 40 40 40 30 30 30



## Exemplo de entrada e saída

11

30 30 30 40 40 40 40 40 30 30 30



## Exemplo de entrada e saída

11

30 30 30 40 40 40 40 40 30 30 30 → 5



## Solução

## Solução

- ★ O problema consiste em identificar as sequências de números consecutivos

## Solução

- ★ O problema consiste em identificar as sequências de números consecutivos
- ★ Uma vez identificadas, a resposta será o tamanho da maior delas



## Solução

- ★ O problema consiste em identificar as sequências de números consecutivos
- ★ Uma vez identificadas, a resposta será o tamanho da maior delas
- ★ Uma possível solução usa um laço e mantém o registro do último elemento visto

## Solução

- ★ O problema consiste em identificar as sequências de números consecutivos
- ★ Uma vez identificadas, a resposta será o tamanho da maior delas
- ★ Uma possível solução usa um laço e mantém o registro do último elemento visto
- ★ Este registro deve ser inicializado com o primeiro elemento do vetor

## Solução

- ★ O problema consiste em identificar as sequências de números consecutivos
- ★ Uma vez identificadas, a resposta será o tamanho da maior delas
- ★ Uma possível solução usa um laço e mantém o registro do último elemento visto
- ★ Este registro deve ser inicializado com o primeiro elemento do vetor
- ★ É preciso fazer uma última verificação, após o encerramento do laço, para verificar a última sequência de números consecutivos

```
#include <stdio.h>
```

```
int main()  
{
```

```
} return 0;
```

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int N;
```

```
    scanf("%d", &N);
```

```
}    return 0;
```

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int N;
```

```
    scanf("%d", &N);
```

```
    int anterior;
```

```
    scanf("%d", &anterior);
```

```
}    return 0;
```

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int N;
```

```
    scanf("%d", &N);
```

```
    int anterior;
```

```
    scanf("%d", &anterior);
```

```
    int consecutivos = 1, resposta = 1;
```

```
}    return 0;
```

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int N;
```

```
    scanf("%d", &N);
```

```
}
```

```
    int anterior;
```

```
    scanf("%d", &anterior);
```

```
    int consecutivos = 1, resposta = 1;
```

```
    for (int i = 1; i < N; ++i)
```

```
    {
```

```
    }    return 0;
```



```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int N;
```

```
    scanf("%d", &N);
```

```
}
```

```
    int anterior;
```

```
    scanf("%d", &anterior);
```

```
    int consecutivos = 1, resposta = 1;
```

```
    for (int i = 1; i < N; ++i)
```

```
    {
```

```
        int proximo;
```

```
        scanf("%d", &proximo);
```

```
    }    return 0;
```

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int N;
```

```
    scanf("%d", &N);
```

```
}
```

```
    int anterior;
```

```
    scanf("%d", &anterior);
```

```
    int consecutivos = 1, resposta = 1;
```

```
    for (int i = 1; i < N; ++i)
```

```
    {
```

```
        int proximo;
```

```
        scanf("%d", &proximo);
```

```
    }    return 0;
```

```
        if (proximo == anterior)
```

```
            ++consecutivos;
```

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int N;
```

```
    scanf("%d", &N);
```

```
    int anterior;
```

```
    scanf("%d", &anterior);
```

```
    int consecutivos = 1, resposta = 1;
```

```
    for (int i = 1; i < N; ++i)
```

```
    {
```

```
        int proximo;
```

```
        scanf("%d", &proximo);
```

```
        if (proximo == anterior)
```

```
            ++consecutivos;
```

```
        else
```

```
        {
```

```
    }
```

```
}
```

```
}    return 0;
```

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int N;
```

```
    scanf("%d", &N);
```

```
    int anterior;
```

```
    scanf("%d", &anterior);
```

```
    int consecutivos = 1, resposta = 1;
```

```
    for (int i = 1; i < N; ++i)
```

```
    {
```

```
        int proximo;
```

```
        scanf("%d", &proximo);
```

```
        if (proximo == anterior)
```

```
            ++consecutivos;
```

```
        else
```

```
        {
```

```
            if (consecutivos > resposta)
```

```
                resposta = consecutivos;
```

```
        }
```

```
    }
```

```
    } return 0;
```

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int N;
```

```
    scanf("%d", &N);
```

```
    int anterior;
```

```
    scanf("%d", &anterior);
```

```
    int consecutivos = 1, resposta = 1;
```

```
    for (int i = 1; i < N; ++i)
```

```
    {
```

```
        int proximo;
```

```
        scanf("%d", &proximo);
```

```
        if (proximo == anterior)
```

```
            ++consecutivos;
```

```
        else
```

```
        {
```

```
            if (consecutivos > resposta)
```

```
                resposta = consecutivos;
```

```
            anterior = proximo;
```

```
            consecutivos = 1;
```

```
        }
```

```
    }
```

```
    return 0;
```

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int N;
```

```
    scanf("%d", &N);
```

```
    int anterior;
```

```
    scanf("%d", &anterior);
```

```
    int consecutivos = 1, resposta = 1;
```

```
    for (int i = 1; i < N; ++i)
```

```
    {
```

```
        int proximo;
```

```
        scanf("%d", &proximo);
```

```
        if (proximo == anterior)
```

```
            ++consecutivos;
```

```
        else
```

```
        {
```

```
            if (consecutivos > resposta)
```

```
                resposta = consecutivos;
```

```
                anterior = proximo;
```

```
                consecutivos = 1;
```

```
        }
```

```
    }
```

```
    if (consecutivos > resposta)
```

```
        resposta = consecutivos;
```

```
    } return 0;
```

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int N;
```

```
    scanf("%d", &N);
```

```
    int anterior;
```

```
    scanf("%d", &anterior);
```

```
    int consecutivos = 1, resposta = 1;
```

```
    for (int i = 1; i < N; ++i)
```

```
    {
```

```
        int proximo;
```

```
        scanf("%d", &proximo);
```

```
        if (proximo == anterior)
```

```
            ++consecutivos;
```

```
        else
```

```
        {
```

```
            if (consecutivos > resposta)
```

```
                resposta = consecutivos;
```

```
            anterior = proximo;
```

```
            consecutivos = 1;
```

```
        }
```

```
    }
```

```
    if (consecutivos > resposta)
```

```
        resposta = consecutivos;
```

```
    printf("%d\n", resposta);
```

```
    } return 0;
```

**Bônus**



## Bônus

- ★ Há uma solução alternativa para este problema

## Bônus

- ★ Há uma solução alternativa para este problema
- ★ Ela é baseada em uma técnica denominada **dois ponteiros**

## Bônus

- ★ Há uma solução alternativa para este problema
- ★ Ela é baseada em uma técnica denominada **dois ponteiros**
- ★ Esta técnica usa duas variáveis inteiras (L e R) para identificar subintervalos [L, R) de elementos com características comuns

## Bônus

- ★ Há uma solução alternativa para este problema
- ★ Ela é baseada em uma técnica denominada **dois ponteiros**
- ★ Esta técnica usa duas variáveis inteiras (L e R) para identificar subintervalos [L, R) de elementos com características comuns
- ★ Como ambas variável apenas avançam, sem retroceder, a eficiência é mesma da solução anterior

## Bônus

- ★ Há uma solução alternativa para este problema
- ★ Ela é baseada em uma técnica denominada **dois ponteiros**
- ★ Esta técnica usa duas variáveis inteiras (L e R) para identificar subintervalos [L, R) de elementos com características comuns
- ★ Como ambas variável apenas avançam, sem retroceder, a eficiência é mesma da solução anterior
- ★ Esta abordagem dispensa o último teste

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int N;
```

```
    scanf("%d", &N);
```

```
        return 0;
```

```
}
```

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int N;
```

```
    scanf("%d", &N);
```

```
    int V[N];
```

```
        return 0;
```

```
}
```

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int N;
```

```
    scanf("%d", &N);
```

```
    int V[N];
```

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

```
        scanf("%d", &V[i]);
```

```
        return 0;
```

```
}
```



```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int N;
```

```
    scanf("%d", &N);
```

```
    int V[N];
```

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

```
        scanf("%d", &V[i]);
```

```
    int resposta = 0, L = 0;
```

```
        return 0;
```

```
}
```

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int N;
```

```
    scanf("%d", &N);
```

```
    int V[N];
```

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

```
        scanf("%d", &V[i]);
```

```
    int resposta = 0, L = 0;
```

```
    while (L < N)
```

```
    {
```

```
}
```

```
    return 0;
```

```
}
```

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int N;
```

```
    scanf("%d", &N);
```

```
    int V[N];
```

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

```
        scanf("%d", &V[i]);
```

```
    int resposta = 0, L = 0;
```

```
    while (L < N)
```

```
    {
```

```
        int R = L + 1;
```

```
}
```

```
    return 0;
```

```
}
```

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int N;
```

```
    scanf("%d", &N);
```

```
    int V[N];
```

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

```
        scanf("%d", &V[i]);
```

```
    int resposta = 0, L = 0;
```

```
    while (L < N)
```

```
    {
```

```
        int R = L + 1;
```

```
        while (R < N && V[R] == V[L])
```

```
            ++R;
```

```
    }
```

```
    return 0;
```

```
}
```

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int N;
```

```
    scanf("%d", &N);
```

```
    int V[N];
```

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

```
        scanf("%d", &V[i]);
```

```
    int resposta = 0, L = 0;
```

```
    while (L < N)
```

```
    {
```

```
        int R = L + 1;
```

```
        while (R < N && V[R] == V[L])
```

```
            ++R;
```

```
        if (R - L > resposta)
```

```
            resposta = R - L;
```

```
    }
```

```
    return 0;
```

```
}
```

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int N;
```

```
    scanf("%d", &N);
```

```
    int V[N];
```

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

```
        scanf("%d", &V[i]);
```

```
    int resposta = 0, L = 0;
```

```
    while (L < N)
```

```
    {
```

```
        int R = L + 1;
```

```
        while (R < N && V[R] == V[L])
```

```
            ++R;
```

```
        if (R - L > resposta)
```

```
            resposta = R - L;
```

```
        L = R;
```

```
    }
```

```
    return 0;
```

```
}
```

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int N;
```

```
    scanf("%d", &N);
```

```
    int V[N];
```

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

```
        scanf("%d", &V[i]);
```

```
    int resposta = 0, L = 0;
```

```
    while (L < N)
```

```
    {
```

```
        int R = L + 1;
```

```
        while (R < N && V[R] == V[L])
```

```
            ++R;
```

```
        if (R - L > resposta)
```

```
            resposta = R - L;
```

```
        L = R;
```

```
    }
```

```
    printf("%d\n", resposta);
```

```
    return 0;
```

```
}
```