OBI 2024 - Nível 2: Fase 1

Concurso

Prof. Edson Alves

Faculdade UnB Gama

Cláudia trabalha na OBI (Organização dos Bons Informáticos), que recentemente realizou um concurso para contratar novos funcionários. Agora, Cláudia tem a tarefa de determinar a *nota de corte* para o concurso. Chamamos de nota de corte a nota mínima necessária para ser aprovado no concurso. Ou seja, se a nota de corte do concurso for C, então todos os participantes com uma nota maior ou igual a C serão aprovados no concurso e todos com nota menor que C serão reprovados.

Seu chefe pediu para que Cláudia aprove no mínimo K candidatos do concurso para a próxima fase, mas ela também não quer que a nota de corte seja muito baixa. Por isso, Cláudia decidiu que a nota de corte deverá ser a maior nota C que faz com que no mínimo K candidatos sejam aprovados.

Sua tarefa é: dados o número N de candidatos, as notas $A_1, A_2, \ldots, A_N$ dos candidatos e a quantidade mínima de aprovados K, diga qual deve ser a maior nota de corte C para que pelo menos K candidatos sejam aprovados.

Entrada

A primeira linha da entrada contém dois inteiros, N e K, representando, respectivamente, o número de participantes e o número mínimo de candidatos que devem ser aprovados.

A segunda linha da entrada contém N inteiros A_i, representando as notas dos participantes.

Saída

Seu programa deve imprimir uma linha contendo um único inteiro C, a nota de corte que deve ser escolhida por Cláudia.

Restrições

- $1 \leq K \leq N \leq 500$
- $1 \leq A_i \leq 100$ para todo $1 \leq i \leq N$

Informações sobre a pontuação

A tarefa vale 100 pontos. Estes pontos estão distribuídos em subtarefas, cada uma com suas **restrições adicionais** às definidas acima.

- **Subtarefa 1 (0 pontos):** Esta subtarefa é composta apenas pelos exemplos mostrados abaixo. Ela não vale pontos, serve apenas para que você verifique se o seu programa imprime o resultado correto para os exemplos.
- **Subtarefa 2 (20 pontos):** $K = 1$.
- **Subtarefa 3 (20 pontos):** $K = 3$.
- **Subtarefa 4 (20 pontos):** $A_i \leq 2$.
- **Subtarefa 5 (40 pontos):** Sem restrições adicionais.

Exemplo de entrada e saída

Exemplo de entrada e saída

```
3 1
```

Exemplo de entrada e saída

3 1

 # de candidatos

Exemplo de entrada e saída

3 1

Exemplo de entrada e saída

```
3 1
92 83 98
```

Exemplo de entrada e saída

```
3 1
92 83 98
```


Nota do candidato 1

Exemplo de entrada e saída

```
3 1
92 83 98
```


Nota do candidato 2

Exemplo de entrada e saída

```
3 1
92 83 98
```

Exemplo de entrada e saída

```
3 1
92 83 98
```

Exemplo de entrada e saída

```
3  1
92  83  98
```

$C = 70$

Exemplo de entrada e saída

```
3  1
92  83  98
```

$C = 70$

Exemplo de entrada e saída

```
3  1
92  83  98
```

$C = 99$

Exemplo de entrada e saída

```
3  1
92  83  98
```

$C = 99$

Exemplo de entrada e saída

```
3  1
92  83  98
```

$C = 90$

Exemplo de entrada e saída

3 1
92 83 98

$C = 90$

Exemplo de entrada e saída

```
3  1
92  83  98
```

$C = 98$

Exemplo de entrada e saída

3 1
92 83 98

$C = 98$

Exemplo de entrada e saída

3 1
92 83 98

$C = 98$

Solução: Subtarefa 2 ($K = 1$)

Solução: Subtarefa 2 ($K = 1$)

- ⋆ Nesta subtarefa, pelo menos um candidato deve ser aprovado

Solução: Subtarefa 2 ($K = 1$)

- Nesta subtarefa, pelo menos um candidato deve ser aprovado
- Seja M a maior nota obtida entre todos os candidatos

Solução: Subtarefa 2 ($K = 1$)

- Nesta subtarefa, pelo menos um candidato deve ser aprovado
- Seja M a maior nota obtida entre todos os candidatos
- Se a nota de corte for igual a M, ao menos o candidato que obteve M será aprovado

Solução: Subtarefa 2 ($K = 1$)

- ⋆ Nesta subtarefa, pelo menos um candidato deve ser aprovado
- ⋆ Seja M a maior nota obtida entre todos os candidatos
- ⋆ Se a nota de corte for igual a M, ao menos o candidato que obteve M será aprovado
- ⋆ Se a nota de corte for maior que M, ninguém será aprovado

Solução: Subtarefa 2 ($K = 1$)

- ⋆ Nesta subtarefa, pelo menos um candidato deve ser aprovado
- ⋆ Seja M a maior nota obtida entre todos os candidatos
- ⋆ Se a nota de corte for igual a M, ao menos o candidato que obteve M será aprovado
- ⋆ Se a nota de corte for maior que M, ninguém será aprovado
- ⋆ Portanto, para esta subtarefa a resposta é a maior nota obtida entre todos os candidatos

```c
#include <stdio.h>

int main()
{









    return 0;
}
```

```c
#include <stdio.h>

int main()
{
    int N, K;
    scanf("%d %d", &N, &K);









    return 0;
}
```

```c
#include <stdio.h>

int main()
{
    int N, K;
    scanf("%d %d", &N, &K);

    int C = 0;









    return 0;
}
```

```c
#include <stdio.h>

int main()
{
    int N, K;
    scanf("%d %d", &N, &K);

    int C = 0;

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




    }



    return 0;
}
```

```c
#include <stdio.h>

int main()
{
    int N, K;
    scanf("%d %d", &N, &K);

    int C = 0;

    for (int i = 0; i < N; ++i)
    {
        int A;
        scanf("%d", &A);



    }



    return 0;
}
```

```c
#include <stdio.h>

int main()
{
    int N, K;
    scanf("%d %d", &N, &K);

    int C = 0;

    for (int i = 0; i < N; ++i)
    {
        int A;
        scanf("%d", &A);

        if (A > C)
            C = A;
    }



    return 0;
}
```

```c
#include <stdio.h>

int main()
{
    int N, K;
    scanf("%d %d", &N, &K);

    int C = 0;

    for (int i = 0; i < N; ++i)
    {
        int A;
        scanf("%d", &A);

        if (A > C)
            C = A;
    }

    printf("%d\n", C);

    return 0;
}
```

Solução: Subtarefa 4 ($A_i \leq 2$)

Solução: Subtarefa 4 ($A_i \leq 2$)

- ⋆ Nesta subtarefa, todos candidatos tiraram ou nota 1 ou nota 2

Solução: Subtarefa 4 ($A_i \leq 2$)

- ⋆ Nesta subtarefa, todos candidatos tiraram ou nota 1 ou nota 2
- ⋆ Só há duas alternativas para a nota de corte: $C = 1$ e $C = 2$

Solução: Subtarefa 4 ($A_i \leq 2$)

- ⋆ Nesta subtarefa, todos candidatos tiraram ou nota 1 ou nota 2
- ⋆ Só há duas alternativas para a nota de corte: $C = 1$ e $C = 2$
- ⋆ Se a nota de corte for igual a 1, todos serão aprovados

Solução: Subtarefa 4 ($A_i \leq 2$)

- ⋆ Nesta subtarefa, todos candidatos tiraram ou nota 1 ou nota 2
- ⋆ Só há duas alternativas para a nota de corte: $C = 1$ e $C = 2$
- ⋆ Se a nota de corte for igual a 1, todos serão aprovados
- ⋆ Se a nota de corte for igual a 2, apenas os candidatos que tiraram 2 serão aprovados

Solução: Subtarefa 4 ($A_i \leq 2$)

⋆ Nesta subtarefa, todos candidatos tiraram ou nota 1 ou nota 2

⋆ Só há duas alternativas para a nota de corte: $C = 1$ e $C = 2$

⋆ Se a nota de corte for igual a 1, todos serão aprovados

⋆ Se a nota de corte for igual a 2, apenas os candidatos que tiraram 2 serão aprovados

⋆ Portanto, a resposta só será 2 quanto o número de candidatos que tiraram 2 for maior ou igual a K; caso contrário, a resposta é igual a 1

```c
#include <stdio.h>

int main()
{









    return 0;
}
```

```c
#include <stdio.h>

int main()
{
    int N, K;
    scanf("%d %d", &N, &K);









    return 0;
}
```

```c
#include <stdio.h>

int main()
{
    int N, K;
    scanf("%d %d", &N, &K);

    int nota2 = 0;









    return 0;
}
```

```c
#include <stdio.h>

int main()
{
    int N, K;
    scanf("%d %d", &N, &K);

    int nota2 = 0;

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




    }



    return 0;
}
```

```c
#include <stdio.h>

int main()
{
    int N, K;
    scanf("%d %d", &N, &K);

    int nota2 = 0;

    for (int i = 0; i < N; ++i)
    {
        int A;
        scanf("%d", &A);



    }



    return 0;
}
```

```c
#include <stdio.h>

int main()
{
    int N, K;
    scanf("%d %d", &N, &K);

    int nota2 = 0;

    for (int i = 0; i < N; ++i)
    {
        int A;
        scanf("%d", &A);

        if (A == 2)
            nota2++;
    }



    return 0;
}
```

```c
#include <stdio.h>

int main()
{
    int N, K;
    scanf("%d %d", &N, &K);

    int nota2 = 0;

    for (int i = 0; i < N; ++i)
    {
        int A;
        scanf("%d", &A);

        if (A == 2)
            nota2++;
    }

    printf("%d\n", nota2 >= K ? 2 : 1);

    return 0;
}
```

Solução: Subtarefa 3 ($K = 3$)

Solução: Subtarefa 3 ($K = 3$)

- ⋆ Nesta subtarefa deve ser classificados, no mínimo, 3 candidatos

Solução: Subtarefa 3 ($K = 3$)

- ⋆ Nesta subtarefa deve ser classificados, no mínimo, 3 candidatos
- ⋆ Na Subtarefa 2 a resposta era a nota do primeiro classificado

Solução: Subtarefa 3 ($K = 3$)

- Nesta subtarefa deve ser classificados, no mínimo, 3 candidatos
- Na Subtarefa 2 a resposta era a nota do primeiro classificado
- Nesta subtarefa, a resposta será a nota do terceiro colocado no concurso

Solução: Subtarefa 3 ($K = 3$)

- ⋆ Nesta subtarefa deve ser classificados, no mínimo, 3 candidatos
- ⋆ Na Subtarefa 2 a resposta era a nota do primeiro classificado
- ⋆ Nesta subtarefa, a resposta será a nota do terceiro colocado no concurso
- ⋆ Para determinar esta nota, é preciso manter o registro das três maiores notas observadas até o momento

Solução: Subtarefa 3 ($K = 3$)

- ⋆ Nesta subtarefa deve ser classificados, no mínimo, 3 candidatos
- ⋆ Na Subtarefa 2 a resposta era a nota do primeiro classificado
- ⋆ Nesta subtarefa, a resposta será a nota do terceiro colocado no concurso
- ⋆ Para determinar esta nota, é preciso manter o registro das três maiores notas observadas até o momento
- ⋆ A cada iteração do laço, as variáveis x, y e z serão atualizadas para que representem a terceira, a segunda e a primeira melhor nota, respectivamente

```c
#include <stdio.h>

int main()
{
    int N, K;
    scanf("%d %d", &N, &K);

    return 0;
}
```

```c
#include <stdio.h>

int main()
{
    int N, K;
    scanf("%d %d", &N, &K);

    int x = 0, y = 0, z = 0;
```

```c
    return 0;
}
```

```c
#include <stdio.h>

int main()
{
    int N, K;
    scanf("%d %d", &N, &K);

    int x = 0, y = 0, z = 0;

    for (int i = 0; i < N; ++i)
    {
        int A;
        scanf("%d", &A);
    }

    return 0;
}
```

```c
#include <stdio.h>

int main()
{
    int N, K;
    scanf("%d %d", &N, &K);

    int x = 0, y = 0, z = 0;

    for (int i = 0; i < N; ++i)
    {
        int A;
        scanf("%d", &A);

        int a = A, b = z;
```

```c
    }

    return 0;
}
```

```c
#include <stdio.h>

int main()
{
    int N, K;
    scanf("%d %d", &N, &K);

    int x = 0, y = 0, z = 0;

    for (int i = 0; i < N; ++i)
    {
        int A;
        scanf("%d", &A);

        int a = A, b = z;

        z = a > b ? a : b;
        a = a < b ? a : b;
```

```c
    }

    return 0;
}
```

```c
#include <stdio.h>

int main()
{
    int N, K;
    scanf("%d %d", &N, &K);

    int x = 0, y = 0, z = 0;

    for (int i = 0; i < N; ++i)
    {
        int A;
        scanf("%d", &A);

        int a = A, b = z;

        z = a > b ? a : b;
        a = a < b ? a : b;

        b = y;
        y = a > b ? a : b;
        a = a < b ? a : b;
    }

    return 0;
}
```

```c
#include <stdio.h>

int main()
{
    int N, K;
    scanf("%d %d", &N, &K);

    int x = 0, y = 0, z = 0;

    for (int i = 0; i < N; ++i)
    {
        int A;
        scanf("%d", &A);

        int a = A, b = z;

        z = a > b ? a : b;
        a = a < b ? a : b;

        b = y;
        y = a > b ? a : b;
        a = a < b ? a : b;

        b = x;
        x = a > b ? a : b;
    }

    return 0;
}
```

```c
#include <stdio.h>

int main()
{
    int N, K;
    scanf("%d %d", &N, &K);

    int x = 0, y = 0, z = 0;

    for (int i = 0; i < N; ++i)
    {
        int A;
        scanf("%d", &A);

        int a = A, b = z;

        z = a > b ? a : b;
        a = a < b ? a : b;

        b = y;
        y = a > b ? a : b;
        a = a < b ? a : b;

        b = x;
        x = a > b ? a : b;
    }

    printf("%d\n", x);

    return 0;
}
```

Solução

Solução

- Para resolver o caso geral do problema, é preciso armazenar todas as notas em um vetor

Solução

⋆ Para resolver o caso geral do problema, é preciso armazenar todas as notas em um vetor

⋆ A nota do i-ésimo candidato será dada por `As[i]`, sendo que a contagem começa em zero

Solução

⋆ Para resolver o caso geral do problema, é preciso armazenar todas as notas em um vetor

⋆ A nota do i-ésimo candidato será dada por `As[i]`, sendo que a contagem começa em zero

⋆ A estratégia de solução será uma busca completa em C

Solução

⋆ Para resolver o caso geral do problema, é preciso armazenar todas as notas em um vetor

⋆ A nota do i-ésimo candidato será dada por `As[i]`, sendo que a contagem começa em zero

⋆ A estratégia de solução será uma busca completa em C

⋆ Iniciando em $C = 100$, serão avaliadas as possíveis notas de corte, em ordem decrescente, até encontrar o primeiro valor para o qual o número de aprovados é maior ou igual a K

Solução

⋆ Para resolver o caso geral do problema, é preciso armazenar todas as notas em um vetor

⋆ A nota do i-ésimo candidato será dada por `As[i]`, sendo que a contagem começa em zero

⋆ A estratégia de solução será uma busca completa em C

⋆ Iniciando em $C = 100$, serão avaliadas as possíveis notas de corte, em ordem decrescente, até encontrar o primeiro valor para o qual o número de aprovados é maior ou igual a K

⋆ Este solução tem complexidade $O(NC)$

```c
#include <stdio.h>

int main()
{
    return 0;
}
```

```c
#include <stdio.h>

int main()
{
    int N, K;
    scanf("%d %d", &N, &K);

    return 0;
}
```

```c
#include <stdio.h>

int main()
{
    int N, K;
    scanf("%d %d", &N, &K);

    int As[N];

    return 0;
}
```

```c
#include <stdio.h>

int main()
{
    int N, K;
    scanf("%d %d", &N, &K);

    int As[N];

    for (int i = 0; i < N; ++i)
        scanf("%d", &As[i]);

    return 0;
}
```

```c
#include <stdio.h>

int main()
{
    int N, K;
    scanf("%d %d", &N, &K);

    int As[N];

    for (int i = 0; i < N; ++i)
        scanf("%d", &As[i]);

    for (int C = 100; C >= 1; --C)
    {
        }

        return 0;
    }
```

```c
#include <stdio.h>

int main()
{
    int N, K;
    scanf("%d %d", &N, &K);

    int As[N];

    for (int i = 0; i < N; ++i)
        scanf("%d", &As[i]);

    for (int C = 100; C >= 1; --C)
    {
        int aprovados = 0;
    }

    return 0;
}
```

```c
#include <stdio.h>

int main()
{
    int N, K;
    scanf("%d %d", &N, &K);

    int As[N];

    for (int i = 0; i < N; ++i)
        scanf("%d", &As[i]);

    for (int C = 100; C >= 1; --C)
    {
        int aprovados = 0;

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


        }
    }

    return 0;
}
```

```c
#include <stdio.h>

int main()
{
    int N, K;
    scanf("%d %d", &N, &K);

    int As[N];

    for (int i = 0; i < N; ++i)
        scanf("%d", &As[i]);

    for (int C = 100; C >= 1; --C)
    {
        int aprovados = 0;

        for (int i = 0; i < N; ++i)
        {
            if (As[i] >= C)
                aprovados++;
        }
        }

        return 0;
    }
```

```c
#include <stdio.h>

int main()
{
    int N, K;
    scanf("%d %d", &N, &K);

    int As[N];

    for (int i = 0; i < N; ++i)
        scanf("%d", &As[i]);

    for (int C = 100; C >= 1; --C)
    {
        int aprovados = 0;

        for (int i = 0; i < N; ++i)
        {
            if (As[i] >= C)
                aprovados++;
        }

        if (aprovados >= K)
        {


        }
    }

    return 0;
}
```

```c
#include <stdio.h>

int main()
{
    int N, K;
    scanf("%d %d", &N, &K);

    int As[N];

    for (int i = 0; i < N; ++i)
        scanf("%d", &As[i]);

    for (int C = 100; C >= 1; --C)
    {
        int aprovados = 0;

        for (int i = 0; i < N; ++i)
        {
            if (As[i] >= C)
                aprovados++;
        }

        if (aprovados >= K)
        {
            printf("%d\n", C);

        }
    }

    return 0;
}
```

```c
#include <stdio.h>

int main()
{
    int N, K;
    scanf("%d %d", &N, &K);

    int As[N];

    for (int i = 0; i < N; ++i)
        scanf("%d", &As[i]);

    for (int C = 100; C >= 1; --C)
    {
        int aprovados = 0;

        for (int i = 0; i < N; ++i)
        {
            if (As[i] >= C)
                aprovados++;
        }

        if (aprovados >= K)
        {
            printf("%d\n", C);
            break;
        }
    }

    return 0;
}
```

Bônus

Bônus

- Há uma solução mais eficiente para este problema

Bônus

- Há uma solução mais eficiente para este problema
- Por meio da ordenação das notas, é possível adotar uma estratégia gulosa

Bônus

- ⋆ Há uma solução mais eficiente para este problema
- ⋆ Por meio da ordenação das notas, é possível adotar uma estratégia gulosa
- ⋆ A resposta será a nota do K-ésimo aprovado

Bônus

⋆ Há uma solução mais eficiente para este problema

⋆ Por meio da ordenação das notas, é possível adotar uma estratégia gulosa

⋆ A resposta será a nota do K-ésimo aprovado

⋆ Se o vetor As for ordenado em ordem crescente, o K-ésimo aprovado ocupará o índice $N - K$ do vetor

Bônus

- ⋆ Há uma solução mais eficiente para este problema
- ⋆ Por meio da ordenação das notas, é possível adotar uma estratégia gulosa
- ⋆ A resposta será a nota do K-ésimo aprovado
- ⋆ Se o vetor As for ordenado em ordem crescente, o K-ésimo aprovado ocupará o índice $N - K$ do vetor
- ⋆ Este solução tem complexidade $O(N \log N)$

```c
#include <stdio.h>


int main()
{









    return 0;
}
```

```c
#include <stdio.h>


int main()
{
    int N, K;
    scanf("%d %d", &N, &K);









    return 0;
}
```

```c
#include <stdio.h>


int main()
{
    int N, K;
    scanf("%d %d", &N, &K);

    int As[N];

    for (int i = 0; i < N; ++i)
        scanf("%d", &As[i]);




    return 0;
}
```

```cpp
#include <stdio.h>


int main()
{
    int N, K;
    scanf("%d %d", &N, &K);

    int As[N];

    for (int i = 0; i < N; ++i)
        scanf("%d", &As[i]);

    std::sort(As, As + N);



    return 0;
}
```

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

int main()
{
    int N, K;
    scanf("%d %d", &N, &K);

    int As[N];

    for (int i = 0; i < N; ++i)
        scanf("%d", &As[i]);

    std::sort(As, As + N);



    return 0;
}
```

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

int main()
{
    int N, K;
    scanf("%d %d", &N, &K);

    int As[N];

    for (int i = 0; i < N; ++i)
        scanf("%d", &As[i]);

    std::sort(As, As + N);

    printf("%d\n", As[N - K]);

    return 0;
}
```