

anotações ITP

soma dos algarismos de um numero

```
#include <iostream>
```

```
using namespace std;
```

```
int main () {
```

```
int N;
```

```
cin >> N;
```

```
int soma = 0;
```

```
while (N >= 1)
```

```
{
```

```
    soma += N % 10;
```

```
    N = N / 10;
```

```
}
```

```
cout << soma << endl;
```

```
return 0;
```

```
}
```

vetor legal

Um vetor é legal caso a diferença entre um elemento e seu antecessor seja par. Escreva um programa que lê um inteiro positivo n e em seguida os n valores do vetor. Então, exibe a mensagem "Legal" quando o vetor for legal, e "Chato" caso contrário. Considere 0 par.

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
int N;
```

```
cin >> N;
```

```
int a[N];
```

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

```
{
```

```
    cin >> a[i];
```

```
}
```

```
// leu entradas
```

```
int chato = 0;
```

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

```
{
```

```
    if((a[i] - a[i-1])% 2 != 0 )
```

```

    {
        chato++;
    }

}
if (chato > 0)
{
    cout << "é chato" << endl;
} else {
    cout << "é legal" << endl;
}

return 0;
}

```

PIPOCA

```

#include <iostream>
using namespace std;

int main()
{
    int N;
    cin >> N;
    int a[N];

    for (int i = 0; i < N; i++)
    {
        cin >> a[i];
    }

    int e = 0, d = 0;
    for (int i = 0; i < N; i++)
    {
        if (a[i] == 2)
        {
            break;
        }
        else if (a[i] == 1)
        {
            e++;
        }
    }

    for (int i = N; i > 0; i--)
    {
        if (a[i] == 2)
        {
            break;
        }
        else if (a[i] == 1)

```

```

        {
            d++;
        }
    }
    if (e == d)
    {
        cout << "tanto faz";
    }
    else if (e < d)
    {
        cout << "esquerda";
    }
    else
    {
        cout << "direita";
    }
    cout << endl;

    return 0;
}

```

LOTES

```

#include <iostream>
using namespace std;

int main () {
    int N;
    cin >> N;

    int lotes[N];
    for (int i = 0; i < N; i++)
    {
        cin >> lotes[i];
    }
    int v=0;

    for (int i = 0; i < N; i++)
    {
        int j=i;
        if (lotes[i] == 2)
        {
            while (lotes[i + 1] == 0)
            {
                v++;
                i++;
            }
            while (lotes[j -1] == 0)

```

```

        {
            v++;
            j--;
        }

    }
}
cout << v << endl;

return 0;
}

```

VAR

```

#include <iostream>
using namespace std;

int main()
{
    int Na[10], Nb[10];

    for (int i = 0; i < 10; i++)
    {
        cin >> Na[i];
    }

    for (int i = 0; i < 10; i++)
    {
        cin >> Nb[i];
    }

    double Pa[10], Pb[10];

    for (int i = 0; i < 10; i++)
    {
        cin >> Pa[i];
    }
    for (int i = 0; i < 10; i++)
    {
        cin >> Pb[i];
    }
    // todos os valores recebidos

    double menorA = Pa[0];
    double maiorB = Pb[0];

    for (int i = 0; i < 10; i++)
    {

```

```

        if (menorA > Pa[i])
        {
            menorA = Pa[i];
        }
        if (maiorB < Pb[i])
        {
            maiorB = Pb[i];
        }
    }
    // todos os maximos atribuidos

    cout << "A: ";
    int testea = 0;
    for (int i = 0; i < 10; i++)
    {
        if (Pa[i] > 0)
        {
            if (Pa[i] > maiorB)
            {
                cout << Na[i] << " ";
                testea++;
            }
        }
    }
    if (testea == 0)
    {
        cout << "sem impedimentos";
    }
    cout << endl;

    cout << "B: ";
    int testeb = 0;
    for (int i = 0; i < 10; i++)
    {
        if (Pb[i] < 0)
        {
            if (Pb[i] < menorA)
            {
                cout << Nb[i] << " ";
                testeb++;
            }
        }
    }
    if (testeb == 0)
    {
        cout << "sem impedimentos";
    }
    cout << endl;

```

```
    return 0;
}
```

BUBBLE SORT

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

int main()
{
    int N;
    cin >> N;

    int a[N];

    for (int i = 0; i < N; i++)
    {
        cin >> a[i];
    }

    // apenas primeiro termo

    for (int i = 0; i < N; i++)
    {
        cout << a[i] << " ";
    }
    cout << endl;
    for (int j = 0; j < N - 1; j++){
        for (int i = 1; i < N - j; i++)
        {
            int temp = a[i - 1];
            if (a[i - 1] > a[i])
            {
                a[i - 1] = a[i];
                a[i] = temp;
            }
        }
    }
    for (int i = 0; i < N; i++)
    {
        cout << a[i] << " ";
    }
    cout << endl;

}
return 0;
```

```
}
```

EXCHANGE SORT

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

int main()
{
    int N;
    cin >> N;

    int a[N];

    for (int i = 0; i < N; i++)
    {
        cin >> a[i];
    }

    // apenas primeiro termo

    for (int i = 0; i < N; i++)
    {
        cout << a[i] << " ";
    }
    cout << endl;

    //resto
    for (int j = 0; j < N - 1 ; j++){
        for (int i = j+1; i < N ; i++)
        {
            int temp = a[j];
            if (a[j] > a[i])
            {
                a[j] = a[i];
                a[i] = temp;
            }
        }
    }
    for (int i = 0; i < N; i++)
    {
        cout << a[i] << " ";
    }
    cout << endl;

}
return 0;
}
```

PONTE

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

int main()
{
    int N;
    cin >> N;

    int pontos[N];

    for (int i = 0; i < N; i++)
    {
        cin >> pontos[i];
    }
    // leu entradas

    int maiorP = 0;
    for (int i = 0; i < N; i++)
    {
        if (maiorP < pontos[i])
        {
            maiorP = pontos[i];
        }
    }

    int inicio = 0, fim = 0;

    for (int i = 0; i < N; i++)
    {
        if (pontos[i] == maiorP)
        {
            inicio = i;
            break;
        }
    }
    for (int i = N; i > 0; i--)
    {
        if (pontos[i] == maiorP)
        {
            fim = i;
            break;
        }
    }
    cout << fim - inicio - 1 << endl;
```



```
    return 0;  
}
```