# Turno 1

## Números univariados

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

// VERSION 1:
//pre: n >= 0
//post: returns true when n is univariate, and false otherwise
bool is_univariate(int n) {
    int last_digit = n%10;
    while (n != 0 and n%10 == last_digit) n = n/10;
    return n == 0;
}

// VERSION 2:
//pre: n >= 0
//post: returns true when n is univariate, and false otherwise
bool is_univariate(int n) {
    int last_digit = n%10;
    while (n != 0) {
        if (last_digit != n%10) return false;
        n = n/10;
    }
    return true;
}

int main() {
    int n;
    cin >> n;
    int position = 1;
    while (n != -1 and not is_univariate(n)) {
        cin >> n;
        ++position;
    }
    if (n != -1) cout << position;
    else cout << 0;
    cout << endl;
}
```

## Tripletas nulas

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

```
int main() {
    int n;
    cin >> n;
    int total = 0;
    for (int i = 0; i < n; ++i) {
        int parcial = 0;
        int m, x, y;
        cin >> m >> x >> y;
        for (int j = 2; j < m; ++j) {
            int z;
            cin >> z;
            if (y == x + z) ++parcial;
            x = y;
            y = z;
        }
        cout << parcial << endl;
        total += parcial;
    }
    cout << "Total: " << total << endl;
}
```

## Reordenando la entrada (1)

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

// Pre: c es una letra minscula
// Post: retorna true si c es una vocal, false en caso contrario
bool es_vocal(char c) {
    return c == 'a' or c == 'e' or c == 'i' or c == 'o' or c == 'u';
}

// Pre: en la entrada hay una secuencia de letras minsculas
// Post: escribe las consonantes en el mismo orden en que
//       aparecen en la entrada seguidas de las consonantes
//       en orden inverso a como aparecen en la entrada
void consonantes_vocales() {
    char c;
    if (cin >> c) {
        bool vocal = es_vocal(c);
        if (not vocal) cout << c;
        consonantes_vocales();
        if (vocal) cout << c;
    }
```

```
}

int main() {
    consonantes_vocales();
    cout << endl;
}
```

# Turno 2

## Primera potencia de dos

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

// VERSION 1
//pre: n >= 0
//post: returns true when n is a power of 2, and false otherwise
bool is_power_2(int n) {
    if (n == 0) return false;
    while (n != 1 and n%2 == 0) n = n/2;
    return n == 1;
}

// VERSION 2:
//pre: n >= 0
//post: returns true when n is a power of 2, and false otherwise
bool is_power_2(int n) {
    if (n == 0) return false;
    while (n != 1) {
        if (n%2 == 1) return false;
        n = n/2;
    }
    return true;
}

int main() {
    int n;
    cin >> n;
    int position = 1;
    while (n != -1 and not is_power_2(n)) {
        cin >> n;
        ++position;
    }
    if (n != -1) cout << position;
    else cout << 0;
    cout << endl;
}
```

## Tripletas residuales

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

```
int main() {
    int m;
    int total = 0;
    while (cin >> m) {
        int parcial = 0;
        int x, y;
        cin >> x >> y;
        for (int j = 2; j < m; ++j) {
            int z;
            cin >> z;
            if (x%y == z) ++parcial;
            x = y;
            y = z;
        }
        cout << parcial << endl;
        total += parcial;
    }
    cout << "Total: " << total << endl;
}
```

## Reordenando la entrada (2)

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

// Pre: c es un carcter
// Post: retorna true si c es una letra, false en caso contrario
bool es_letra(char c) {
    return ('a' <= c and c <= 'z') or ('A' <= c and c <= 'Z');
}

// Pre: en la entrada hay una secuencia de caracteres
// Post: escribe los mismos caracteres que en la entrada
//       reordenados: primero los que no son letras (en el
//       mismo orden en el que aparecieron) y, a continuacin,
//       los caracteres que son letras (en orden inverso al de
//       su aparicin)
void signos_letras() {
    char c;
    if (cin >> c) {
        bool letra = es_letra(c);
        if (not letra) cout << c;
        signos_letras();
        if (letra) cout << c;
```

```
    }
}

int main() {
    signos_letras();
    cout << endl;
}
```