

# Programação Imperativa

1º Ano – LCC/MIEF/MIEI

Questões 1ª Parte

1. Defina um programa que lê (usando a função `scanf` uma sequência de números inteiros terminada com o número 0 e imprime no ecran o maior elemento da sequência.  
*Define a program that reads (using the `scanf` function) a sequence of integers ending with the number 0 and prints the largest element of the sequence on the screen.*
2. Defina um programa que lê (usando a função `scanf` uma sequência de números inteiros terminada com o número 0 e imprime no ecran a média da sequência.  
*Define a program that reads (using the `scanf` function) a sequence of integers ending with the number 0 and prints the average of the sequence on the screen.*
3. Defina um programa que lê (usando a função `scanf` uma sequência de números inteiros terminada com o número 0 e imprime no ecran o segundo maior elemento.  
*Define a program that reads (using the `scanf` function) a sequence of integers ending with the number 0 and prints the second largest element on the screen.*
4. Defina uma função `int bitsUm (unsigned int n)` que calcula o número de bits iguais a 1 usados na representação binária de um dado número `n`.  
*Define a function `int bitsOne (unsigned int n)` that calculates the number of bits equal to 1 used in the binary representation of a given number `n`.*  
(<https://codeboard.io/projects/13548/summary>)
5. Defina uma função `int trailingZ (unsigned int n)` que calcula o número de bits a 0 no final da representação binária de um número (i.e., o expoente da maior potência de 2 que é divisor desse número).  
*Define a function `int trailingZ (unsigned int n)` that calculates the number of 0's at the end of the binary representation of a number (i.e., the exponent of the highest power of 2 that is divisible by that number).*  
(<https://codeboard.io/projects/13549/summary>)
6. Defina uma função `int qDig (unsigned int n)` que calcula o número de dígitos necessários para escrever o inteiro `n` em base decimal. Por exemplo, `qDig (440)` deve retornar 3.  
*Define a function `int qDig (unsigned int n)` that calculates the number of digits needed to write the integer `n` in base 10. For example, `qDig (440)` should return 3.*  
(<https://codeboard.io/projects/13583/summary>)
7. Apresente uma definição da função pré-definida em C `char *strcat (char s1[], char s2[])` que concatena a string `s2` a `s1` (retornando o endereço da primeira).

Give a definition of the predefined C function `char *strcat (char s1[], char s2[])` which concatenates the string `s2` to `s1` (returning the address of the first).  
(<https://codeboard.io/projects/14490/summary>)

8. Apresente uma definição da função pré-definida em C `char *strcpy (char *dest, char source[])` que copia a string `source` para `dest` retornando o valor desta última.

Give a definition of the predefined C function `char *strcpy (char *dest, char source[])` that copies the string `source` to `dest` and returns the value of the latter.  
(<https://codeboard.io/projects/14491/summary>)

9. Apresente uma definição da função pré-definida em C `int strcmp (char s1[], char s2[])` que compara (lexicograficamente) duas strings. O resultado deverá ser

- 0 se as strings forem iguais
- <0 se `s1 < s2`
- >0 se `s1 > s2`

Give a definition of the predefined C function `int strcmp (char s1[], char s2[])` that compares (lexicographically) two strings. The result should be

- 0 if the strings are equal
- <0 if `s1 < s2`
- >0 if `s1 > s2`

(<https://codeboard.io/projects/14492/summary>)

10. Apresente uma definição da função pré-definida em C `char *strstr (char s1[], char s2[])` que determina a posição onde a string `s2` ocorre em `s1`. A função deverá retornar NULL caso `s2` não ocorra em `s1`.

Give a definition of the predefined C function `char *strstr (char s1[], char s2[])` that determines the position where the string `s2` occurs in `s1`. The function should return NULL if `s2` does not occur in `s1`.

(<https://codeboard.io/projects/14493/summary>)

11. Defina uma função `void strrev (char s[])` que inverte uma string.

Define a function `void strrev (char s[])` that reverses a string.

(<https://codeboard.io/projects/14494/summary>)

12. Defina uma função `void strnoV (char s[])` que retira todas as vogais de uma string.

Define a function `void strnoV (char s[])` that removes all vowels from a string.

(<https://codeboard.io/projects/13661/summary>)

13. Defina uma função `void truncW (char t[], int n)` que dado um texto `t` com várias palavras (as palavras estão separadas em `t` por um ou mais espaços) e um inteiro `n`, *trunca* todas as palavras de forma a terem no máximo `n` caracteres. Por exemplo, se a string `txt` contiver "liberdade, igualdade e fraternidade", a invocação de `truncW (txt, 4)` deve fazer com que passe a estar lá armazenada a string "libe igua e frat".

Define a function `void truncW (char t[], int n)` which, given a text `t` with several words (the

words are separated in *t* by one or more spaces) and an integer *n*, truncates all the words so that they have at most *n* characters. For example, if string *txt* contains "liberdade, igualdade e fraternidade", invoking *truncW (txt, 4)* should cause the string "libe igua e frat" to be stored in *txt*.

(<https://codeboard.io/projects/13659/summary>)

14. Defina uma função `char charMaisfreq (char s[])` que determina qual o caracter mais frequente numa string. A função deverá retornar 0 no caso de *s* ser a string vazia.

*Define a function char charMaisfreq (char s[]) that computes the most frequent character in a string. The function should return 0 if s is an empty string.*

(<https://codeboard.io/projects/14577>)

15. Defina uma função `int iguaisConsecutivos (char s[])` que, dada uma string *s* calcula o comprimento da maior sub-string com caracteres iguais. Por exemplo, `iguaisConsecutivos ("ababaabcccaac")` deve dar como resultado 3, correspondendo à repetição "ccc".

*Define a function int iguaisConsecutivos (char s[]) which, given a string s, calculates the length of the longest sub-string with equal characters. For example, iguaisConsecutivos ("ababaabcccaac") should result in 3, corresponding to the repetition "ccc".*

(<https://codeboard.io/projects/14578>)

16. Defina uma função `int difConsecutivos (char s[])` que, dada uma string *s* calcula o comprimento da maior sub-string com caracteres diferentes. Por exemplo, `difConsecutivos ("ababcaccaac")` deve dar como resultado 3, correspondendo (por exemplo) à string "abc".

*Define a function int difConsecutivos (char s[]) which, given a string s, calculates the length of the longest sub-string with different characters. For example, difConsecutivos ("ababcaccaac") should result in 3, corresponding to (for example) the string "abc".*

(<https://codeboard.io/projects/14579>)

17. Defina uma função `int maiorPrefixo (char s1 [], char s2 [])` que calcula o comprimento do maior prefixo comum entre as duas strings.

*Define a function int maiorPrefixo (char s1 [], char s2 []) that calculates the length of the longest common prefix between the two strings.*

(<https://codeboard.io/projects/14580>)

18. Defina uma função `int maiorSufixo (char s1 [], char s2 [])` que calcula o comprimento do maior sufixo comum entre as duas strings.

*Define a function int maiorSufixo (char s1 [], char s2 []) that calculates the length of the longest common suffix between the two strings.*

(<https://codeboard.io/projects/14581>)

19. Defina a função `int sufPref (char s1[], char s2[])` que calcula o tamanho do maior sufixo de *s1* que é um prefixo de *s2*. Por exemplo `sufPref("batota","totalidade")` deve dar como resultado 4, uma vez que a string "tota" é um sufixo de "batota" e um prefixo de "totalidade".

*Define the function int sufPref (char s1[], char s2[]) which calculates the length of the largest suffix of s1 which is a prefix of s2. For example, sufPref("batota","totalidade") should result in 4, since the string "tota" is a suffix of "batota" and a prefix of "totalidade".*

(<https://codeboard.io/projects/14582>)

20. Defina uma função `int contaPal (char s[])` que conta as palavras de uma string. Uma palavra é uma sequência de caracteres (diferentes de espaço) terminada por um ou mais espaços. Assim se a string `p` tiver o valor `"a a bb a"`, o resultado de `contaPal (p)` deve ser 4.
- Define a function `int contaPal (char s[])` that counts the words in a string. A word is a sequence of characters (other than spaces) terminated by one or more spaces. If the string `p` has the value `"a a bb a"`, the result of `contaPal (p)` should be 4.*
- (<https://codeboard.io/projects/14583>)
21. Defina uma função `int contaVogais (char s[])` que retorna o número de vogais da string `s`. Não se esqueça de considerar tanto maiúsculas como minúsculas.
- Define a function `int contaVogais (char s[])` that returns the number of vowels in the string `s`. Don't forget to consider both upper and lower case.*
- (<https://codeboard.io/projects/14585>)
22. Defina uma função `int contida (char a[], char b[])` que testa se todos os caracteres da primeira string também aparecem na segunda. Por exemplo, `contida ("braga","bracara augusta")` deve retornar *verdadeiro* enquanto que `contida ("braga","bracarense")` deve retornar *falso*.
- Define a function `int contida (char a[], char b[])` that tests whether all the characters in the first string also appear in the second. For example, `contida ("braga", "bracara augusta")` should return true whereas `contida ("braga", "bracarense")` should return false.*
- (<https://codeboard.io/projects/14586>)
23. Defina uma função `int palindorome (char s[])` que testa se uma palavra é palíndrome, i.e., lê-se de igual forma nos dois sentidos.
- Define a function `int palindorome (char s[])` that tests whether a word is palindromic, i.e. reads the same in both directions.*
- (<https://codeboard.io/projects/14587>)
24. Defina uma função `int remRep (char x[])` que elimina de uma string todos os caracteres que se repetem sucessivamente deixando lá apenas uma cópia. A função deverá retornar o comprimento da string resultante. Assim, por exemplo, ao invocarmos a função com uma vector contendo `"aaabaaabbbbaa"`, o vector deve passar a conter a string `"ababa"` e a função deverá retornar o valor 5.
- Define a function `int remRep (char x[])` that removes from a string all characters that are repeated in succession, leaving only one copy. The function should return the length of the resulting string. So, for example, after we call the function with a vector containing `"aaabaaabbbbaa"`, the vector should now contain the string `"ababa"` and the function should return the value 5.*
- (<https://codeboard.io/projects/13663>)
25. Defina uma função `int limpaEspacos (char t[])` que elimina repetições sucessivas de espaços por um único espaço. A função deve retornar o comprimento da string resultante.
- Define a function `int limpaEspacos (char t[])` that eliminates successive repetitions of spaces by a single space. The function should return the length of the resulting string.*
- (<https://codeboard.io/projects/13733>)
26. Defina uma função `void insere (int v[], int N, int x)` que insere um elemento (`x`) num vector ordenado. Assuma que as `N` primeiras posições do vector estão ordenadas e que por

isso, após a inserção o vector terá as primeiras  $N+1$  posições ordenadas.

*Define a function `void insere (int v[], int N, int x)` that inserts an element ( $x$ ) into an ordered vector. Assume that the first  $N$  positions of the vector are sorted and that therefore, after insertion, the vector will have the first  $N+1$  positions sorted.*

(<https://codeboard.io/projects/14836>)

27. Defina uma função `void merge (int r [], int a[], int b[], int na, int nb)` que, dados vectores ordenados **a** (com **na** elementos) e **b** (com **nb** elementos), preenche o vector **r** (com **na+nb** elementos) com os elementos de **a** e **b** ordenados.

*Define a function `void merge (int r [], int a[], int b[], int na, int nb)` which, given ordered vectors **a** (with **na** elements) and **b** (with **nb** elements), fills the vector **r** (with **na+nb** elements) with the elements of **a** and **b** sorted.*

(<https://codeboard.io/projects/14837>)

28. Defina uma função `int crescente (int a[], int i, int j)` que testa se os elementos do vector **a**, entre as posições **i** e **j** (inclusivé) estão ordenados por ordem crescente. A função deve retornar 1 ou 0 consoante o vector esteja ou não ordenado.

*Define a function `int crescente (int a[], int i, int j)` that tests whether the elements of the vector **a**, between positions **i** and **j** (inclusive) are sorted in increasing order. The function should return 1 or 0 depending on whether the vector is sorted or not.*

(<https://codeboard.io/projects/14838>)

29. Defina uma função `int retiraNeg (int v[], int N)` que retira os números negativos de um vector com **N** inteiros. A função deve retornar o número de elementos que não foram retirados.

*Define a function `int retiraNeg (int v[], int N)` that removes the negative numbers from a vector with **N** integers. The function should return the number of elements that were not removed.*

(<https://codeboard.io/projects/14839>)

30. Defina uma função `int menosFreq (int v[], int N)` que recebe um vector **v** com **N** elementos **ordenado por ordem crescente** e retorna **o menos frequente** dos elementos do vector. Se houver mais do que um elemento nessas condições deve retornar o que começa por aparecer no índice mais baixo.

*Define a function `int menosFreq (int v[], int N)` that receives a vector **v** with **N** elements **ordered in ascending order** and returns **the least frequent** of the elements in the vector. If there is more than one element under these conditions, it should return the one that starts at the lowest index.*

(<https://codeboard.io/projects/14840>)

31. Defina uma função `int maisFreq (int v[], int N)` que recebe um vector **v** com **N** elementos **ordenado por ordem crescente** e retorna **o mais frequente** dos elementos do vector. Se houver mais do que um elemento nessas condições deve retornar o que começa por aparecer no índice mais baixo.

*Define a function `int maisFreq (int v[], int N)` that receives a vector **v** with **N** elements **ordered in ascending order** and returns **the most frequent** of the elements in the vector. If there is more than one element under these conditions, it should return the one that starts at the lowest index.*

(<https://codeboard.io/projects/14841>)

32. Defina uma função `int maxCresc (int v[], int N)` que calcula o comprimento da maior sequência crescente de elementos consecutivos num vector **v** com **N** elementos. Por exemplo,

se o vector contiver 10 elementos pela seguinte ordem: 1, 2, 3, 2, 1, 4, 10, 12, 5, 4, a função deverá retornar 4, correspondendo ao tamanho da sequência 1, 4, 10, 12.

*Define a function `int maxCresc (int v[], int N)` that calculates the length of the longest increasing sequence of consecutive elements in a vector `v` with `N` elements. For example, if the vector contains 10 elements in the following order: 1, 2, 3, 2, 1, 4, 10, 12, 5, 4, the function should return 4, corresponding to the length of the sequence 1, 4, 10, 12.*

(<https://codeboard.io/projects/14842>)

33. Defina uma função `int elimRep (int v[], int n)` que recebe um vector `v` com `n` inteiros e elimina as repetições. A função deverá retornar o número de elementos do vector resultante. Por exemplo, se o vector `v` contiver nas suas primeiras 10 posições os números {1, 2, 3, 2, 1, 4, 2, 4, 5, 4}, a invocação `elimRep (v,10)` deverá retornar 5 e colocar nas primeiras 5 posições do vector os elementos {1,2,3,4,5}.

*Define a function `int elimRep (int v[], int n)` that receives a vector `v` with `n` integers and eliminates the repetitions. The function should return the number of elements in the resulting vector. For example, if the vector `v` contains in its first 10 positions the numbers {1, 2, 3, 2, 1, 4, 2, 4, 5, 4}, the invocation `elimRep (v,10)` should return 5 and place the elements {1,2,3,4,5} in the first 5 positions of the vector.*

(<https://codeboard.io/projects/14843>)

34. Defina uma função `int elimRepOrd (int v[], int n)` que recebe um vector `v` com `n` inteiros ordenado por ordem crescente e elimina as repetições. A função deverá retornar o número de elementos do vector resultante.

*Define a function `int elimRepOrd (int v[], int n)` that receives a vector `v` with `n` integers sorted in ascending order and eliminates repetitions. The function should return the number of elements in the resulting vector.*

(<https://codeboard.io/projects/14844>)

35. Defina uma função `int comunsOrd (int a[], int na, int b[], int nb)` que calcula quantos elementos os vectores `a` (com `na` elementos) e `b` (com `nb` elementos) têm em comum. Assuma que os vectores `a` e `b` estão ordenados por ordem crescente.

*Define a function `int comunsOrd (int a[], int na, int b[], int nb)` that calculates how many elements the vectors `a` (with `na` elements) and `b` (with `nb` elements) have in common. Assume that the vectors `a` and `b` are sorted in ascending order.*

(<https://codeboard.io/projects/14845>)

36. Defina uma função `int comuns (int a[], int na, int b[], int nb)` que calcula quantos elementos os vectores `a` (com `na` elementos) e `b` (com `nb` elementos) têm em comum. Assuma que os vectores `a` e `b` **não estão** ordenados e defina a função sem alterar os vectores.

*Define a function `int comuns (int a[], int na, int b[], int nb)` that calculates how many elements the vectors `a` (with `na` elements) and `b` (with `nb` elements) have in common. Assume that the vectors `a` and `b` are **not** ordered and define the function without changing the vectors.*

(<https://codeboard.io/projects/14846>)

37. Defina uma função `int minInd (int v[], int n)` que, dado um vector `v` com `n` inteiros, retorna o índice do menor elemento do vector.

*Define a function `int minInd (int v[], int n)` which, given a vector `v` with `n` integers, returns the index of the smallest element in the vector.*

(<https://codeboard.io/projects/14847>)

38. Defina uma função `void somasAc (int v[], int Ac [], int N)` que preenche o vector `Ac` com as somas acumuladas do vector `v`. Por exemplo, na posição `Ac[3]` deve ser calculado como `v[0]+v[1]+v[2]+v[3]`.

*Define a function `void somasAc (int v[], int Ac [], int N)` that fills the vector `Ac` with the accumulated sums of the vector `v`. For example, at position `Ac[3]` it should be calculated as `v[0]+v[1]+v[2]+v[3]`.*

(<https://codeboard.io/projects/14848>)

39. Defina uma função `int triSup (int N, float m [N][N])` que testa se uma matriz quadrada é triangular superior, i.e., que todos os elementos abaixo da diagonal são zeros.

*Define a function `int triSup (int N, float m [N][N])` that tests whether a square matrix is upper triangular, i.e., that all elements below the diagonal are zeros.*

(<https://codeboard.io/projects/14849>)

40. Defina uma função `void transposta (int N, float m [N][N])` que transforma uma matriz na sua transposta.

*Define a function `void transposta (int N, float m [N][N])` that transforms a matrix into its transpose.*

(<https://codeboard.io/projects/14850>)

41. Defina uma função `void addTo (int N, int M, int a [N][M], int b[N][M])` que adiciona a segunda matriz à primeira.

*Define a function `void addTo (int N, int M, int a [N][M], int b[N][M])` that adds the second matrix to the first.*

(<https://codeboard.io/projects/14851>)

42. Uma forma de representar conjuntos de índices consiste em usar um array de inteiros contendo 1 ou 0 consoante esse índice pertença ou não ao conjunto. Assim o conjunto  $\{1, 4, 7\}$  seria representado por um array em que as primeiras 8 posições conteriam  $\{0, 1, 0, 0, 1, 0, 0, 1\}$ . Apresente uma definição da função `int unionSet (int N, int v1[N], int v2[N], int r[N])` que coloca no array `r` o resultado da união dos conjuntos `v1` e `v2`.

*One way of representing sets of indexes is to use an array of integers containing 1 or 0 depending on whether that index belongs to the set or not. So the set  $\{1, 4, 7\}$  would be represented by an array in which the first 8 positions would contain  $\{0, 1, 0, 0, 1, 0, 0, 1\}$ .*

*Give a definition of the function `int unionSet (int N, int v1[N], int v2[N], int r[N])` that puts the result of the union of the sets `v1` and `v2` into the array `r`.*

(<https://codeboard.io/projects/14685>)

43. Uma forma de representar conjuntos de índices consiste em usar um array de inteiros contendo 1 ou 0 consoante esse índice pertença ou não ao conjunto. Assim o conjunto  $\{1, 4, 7\}$  seria representado por um array em que as primeiras 8 posições conteriam  $\{0, 1, 0, 0, 1, 0, 0, 1\}$ . Apresente uma definição da função `int intersectSet (int N, int v1[N], int v2[N], int r[N])` que coloca no array `r` o resultado da intersecção dos conjuntos `v1` e `v2`.

*One way of representing sets of indexes is to use an array of integers containing 1 or 0 depending on whether that index belongs to the set or not. So the set  $\{1, 4, 7\}$  would be represented by an array in which the first 8 positions would contain  $\{0, 1, 0, 0, 1, 0, 0, 1\}$ .*

*Give a definition of the function `int intersectSet (int N, int v1[N], int v2[N], int r[N])` that puts the result of the intersection of the sets `v1` and `v2` into the array `r`.*

(<https://codeboard.io/projects/14694>)

44. Uma forma de representar multi-conjuntos de índices consiste em usar um array de inteiros contendo em cada posição o número de ocorrências desse índice. Assim o multi-conjunto  $\{1, 1, 4, 7, 7, 7\}$  seria representado por um array em que as primeiras 8 posições conteriam  $\{0, 2, 0, 0, 1, 0, 0, 3\}$ .

Apresente uma definição da função `int intersectMSet (int N, int v1[N], int v2[N], int r[N])` que coloca no array `r` o resultado da intersecção dos multi-conjuntos `v1` e `v2`.

*One way of representing multi-sets of indexes is to use an array of integers containing in each position the number of occurrences of that index. So the multi-set  $\{1, 1, 4, 7, 7\}$  would be represented by an array in which the first 8 positions would contain  $\{0, 2, 0, 0, 1, 0, 0, 3\}$ .*

*Give a definition of the function `int intersectMSet (int N, int v1[N], int v2[N], int r[N])` that puts the result of the intersection of the multi-sets `v1` and `v2` into the array `r`.*

(<https://codeboard.io/projects/14733>)

45. Uma forma de representar multi-conjuntos de índices consiste em usar um array de inteiros contendo em cada posição o número de ocorrências desse índice. Assim o multi-conjunto  $\{1, 1, 4, 7, 7, 7\}$  seria representado por um array em que as primeiras 8 posições conteriam  $\{0, 2, 0, 0, 1, 0, 0, 3\}$ .

Apresente uma definição da função `int unionMSet (int N, int v1[N], int v2[N], int r[N])` que coloca no array `r` o resultado da união dos multi-conjuntos `v1` e `v2`.

*One way of representing multi-sets of indexes is to use an array of integers containing in each position the number of occurrences of that index. So the multi-set  $\{1, 1, 4, 7, 7\}$  would be represented by an array in which the first 8 positions would contain  $\{0, 2, 0, 0, 1, 0, 0, 3\}$ .*

*Give a definition of the function `int unionMSet (int N, int v1[N], int v2[N], int r[N])` that puts the result of the union of the multi-sets `v1` and `v2` into the array `r`.*

(<https://codeboard.io/projects/14734>)

46. Uma forma de representar multi-conjuntos de índices consiste em usar um array de inteiros contendo em cada posição o número de ocorrências desse índice. Assim o multi-conjunto  $\{1, 1, 4, 7, 7, 7\}$  seria representado por um array em que as primeiras 8 posições conteriam  $\{0, 2, 0, 0, 1, 0, 0, 3\}$ .

Apresente uma definição da função `int cardinalMSet (int N, int v[N])` que calcula o número de elementos do multi-conjunto `v`.

*One way of representing multi-sets of indexes is to use an array of integers containing in each position the number of occurrences of that index. So the multi-set  $\{1, 1, 4, 7, 7\}$  would be represented by an array in which the first 8 positions would contain  $\{0, 2, 0, 0, 1, 0, 0, 3\}$ .*

*Give a definition of the function `int cardinalMSet (int N, int v[N])` which calculates the number of elements in the multiset `v`.*

(<https://codeboard.io/projects/14740>)

47. Considere as seguintes definições para representar as posições e movimentos de um robot.

```
typedef enum movimento {Norte, Oeste, Sul, Este} Movimento;
typedef struct posicao {
    int x, y;
} Posicao;
```

Defina a função `Posicao posFinal (Posicao inicial, Movimento mov[], int N)` que, dada uma posição inicial e um array com `N` movimentos, calcula a posição final do robot



depois de efectuar essa sequência de movimentos.

*Consider the definitions above to represent the positions and movements of a robot.*

*Define the function `Position posFinal (Posicao inicial, Movimento mov[], int N)` which, given an initial position and an array with  $N$  movements, calculates the robot's final position after performing this sequence of movements.*

(<https://codeboard.io/projects/73018>)

48. Considere as seguintes definições para representar as posições e movimentos de um robot.

```
typedef enum movimento {Norte, Oeste, Sul, Este} Movimento;
typedef struct posicao {
    int x, y;
} Posicao;
```

Defina a função `int caminho (Posicao inicial, Posicao final, Movimento mov[], int N)` que, dadas as posições inicial e final do robot, preenche o array com os movimentos suficientes para que o robot passe de uma posição para a outra.

A função deverá preencher no máximo  $N$  elementos do array e retornar o número de elementos preenchidos. Se não for possível atingir a posição final com  $N$  movimentos, a função deverá retornar um número negativo.

*Consider the definitions above to represent the positions and movements of a robot.*

*Define the function `int caminho (Posicao inicial, Posicao final, Movimento mov[], int N)` which, given the robot's start and end positions, fills the array with enough movements for the robot to move from one position to the other.*

*The function should fill a maximum of  $N$  elements of the array and return the number of elements filled. If it is not possible to reach the final position with  $N$  moves, the function should return a negative number.*

(<https://codeboard.io/projects/73019>)

49. Considere o seguinte tipo para representar a posição de um robot numa grelha.

```
typedef struct posicao {
    int x, y;
} Posicao;
```

Defina a função `int maisCentral (Posicao pos[], int N)` que, dado um array com  $N$  posições, determina o índice da posição que está mais perto da origem (note que as coordenadas de cada ponto são números inteiros).

*Consider the above definition to represent the position of a robot.*

*Define the function `int maisCentral (Posicao pos[], int N)` which, given an array with  $N$  positions, determines the index of the position that is closest to the origin (note that the coordinates of each point are integers).*

(<https://codeboard.io/projects/73020>)

50. Considere o seguinte tipo para representar a posição de um robot numa grelha.

```
typedef struct posicao {
```

```
        int x, y;  
    } Posicao;
```

Defina a função `int vizinhos (Posicao p, Posicao pos[], int N)` que, dada uma posição e um array com  $N$  posições, calcula quantas dessas posições são adjacentes à posição dada.

*Consider the definition above to represent the position of a robot.*

*Define the function `int vizinhos (Posicao p, Posicao pos[], int N)` which, given a position and an array with  $N$  positions, calculates how many of these positions are adjacent to the given position.*

(<https://codeboard.io/projects/73021>)