
Imperative Programming 2024/2025

Model Exam

Duration: 2h

Student ID:

Full name: _____

Group 1 - Fundamentals of C (20%)

1.1. Write small **portions of code** for each of the following items:

a) Extract the hundreds digit of an integer n :

b) Obtain the absolute value of an integer n :

c) What does this code fragment output?

```
int b = 2;
int a = b*b + 1;
printf("%d %d\n", a, b);
```

d) Boolean expression to verify if an integer is odd:

e) Count the number of digits of a positive integer n :

f) Boolean expression to verify if a char c is a decimal digit:

g) What is the result of calling `fib(5)`?

```
int fib(int n) {
    if (n==0 || n==1)
        return 1;
    else
        return fib(n-1) + fib(n-2);
}
```

h) How do we write the last letter of a string s ?

i) How can we fill the main diagonal of matrix `int m[4][4]` with value 22?

j) Consider `p` as a pointer to an integer. Write an expression to increment by one the integer pointed by `p`.

k) If `p` is a pointer to a `struct Person`, how can we output its attribute `name`?

1) Consider the following incomplete function to search, using binary search, for a value x in an indexed variable `vec[]` that is ordered and has size n . Provide the code for instructions A and B.

```
int pesquisa_bin(int vec[], int n, int x) {
    int i = 0, j = n-1;
    while (i <= j) {
        int k = (i+j)/2;    // middle index
        if (vec[k] == x)
            return k;      // found
        else if (x > vec[k])
            ???;            // instruction A
        else
            ???;            // instruction B
    }
    return -1;             // not found
}
```

Group 2 - Complexity of programs (10%)

2.1. By considering the worst case scenario in terms of execution time how would you classify the complexity of the following code fragments (use terms such as constant, logarithmic, linear, linearithmic, quadratic, exponential).

a)

```
for (int i=0; i<n; i++)
    for (int j=0; j<i; j++) count++;
```

b) `for (int j=0; j<100; j++) count++;`

c) `for (int i=n; i>0; i=i/2) count++;`

2.2. Consider an implementation of singly linked lists in which the values of the elements (or nodes) are words and the sequence isn't ordered. Be n the number of words in the list (equal to the number of elements). Say what is the expected complexity of the following operations (justify):

a) return the first word of the sequence:

b) Verify if the sequence contains a given word:

Group 3 - Sorting and Recursion (10%)

3.1. The binomial coefficients of the Pascal triangle are given by the following recurrence relations:

$$C(n, 0) = 0 \quad e \quad C(n, n) = 1 \quad para \quad n \geq 0.$$

$$C(n, k) = C(n - 1, k) + C(n - 1, k - 1) \quad para \quad n > k > 0.$$

Write a recursive function to generate $C(n, k)$ by the given relation.

3.2. Consider the following data type:

```
typedef struct {
    char *name; // string com o nome
    int age; // idade
    ... // outros atributos
} Person;
```

Write a function that compares two variables of type **Person** by age (increasing order). If they have the same age, the order is decided alphabetically.

Group 4 - Structures (30%)

4.1. Consider the following structure `Matrix` to represent a matrix of integers.

```
typedef struct {  
    int **data; // dynamic array  
    int rows;   // number of rows  
    int cols;   // number of columns  
} Matrix;
```

Write a function `Matrix initMatrix(int rows, int cols, int k)` that initializes the matrix of size `rows` x `cols` with a value `k` in all positions.

4.2. Write a function `void setValue(Matrix *m, int row, int col, int v)` that modifies the value in position `(row, col)` of `m` to `v`.

4.3. Write a function `void largestSum(Matrix m, int *sumMax, int *rowMax)` that returns in `sumMax` the highest sum of the elements of each row of the matrix `m` and in `rowMax` returns the row where that sum occurs. You can assume that all values in the matrix are greater or equal 0.

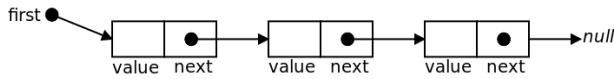
4.4. Write a function `Matrix transpose(Matrix m)` that receives a matrix m and returns its transpose (i.e., changes rows by columns). For example, the following figure shows a matrix and its transpose:

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{bmatrix}$$

(you can use the remaining space to continue answers from this group if you need more space)

Group 5 - Lists, Stacks and Queues (30%)

5.1. Consider the implementation of linked lists as given in classes, typed `LinkedList`, with attributes `size` and `first`, and a structure type `Node` to represent a node of the list with attributes `value` and `next`.



a) Write a function `bool contains(LinkedList *l, NodeInfo x)` that returns *true* if element *x* is found in the list and *false* otherwise. You can't use other methods from classes.

b) Write a function `void remove(LinkedList *l, int i)` that transforms the list by removing the *i*th element (element in position *i*). The positions start in 0. If position *i* doesn't exist, the function does nothing. You can't use other methods from classes.

c) Start by explaining which functions would you use in order to define a stack based on the implementation of linked lists used in this exercise. Consider that `NodeInfo` is an integer and exemplify by writing code instructions that allow the declaration of a variable `s` as a stack, add 5 to the stack, add 7 to the stack, remove and write the value at the top of the stack. Which value is written?

Student ID

--	--	--	--	--	--	--	--	--

Name:

(you can use this page as a continuation page if needed)