**Laborator 15**

**În REPORT.txt adăugați output-ul versiunii finale a programului. Dacă o parte din program nu e implementată, nu funcționează, face ca programul să dea seg fault atunci puteți comenta unele linii din main și să folosiți aceea afișare.**

**Exerciții**

1. (**hashMap.c**) Pentru un HashMap implementați funcția de inserție. **insertKeyValue()**
2. (**hashMap.c**) Pentru un HashMap implementați funcția de extragere a unei valori. **getValue()**
3. (**hashMap.c**) Pentru un HashMap implementați funcția de ștergere a unei valori. **removeKey()**
4. (**binarySearch.c**) Implementați algoritmul de [căutare binară](https://en.wikipedia.org/wiki/Binary_search_algorithm). **binarySearch ()**
5. (**bubbleSort.c**) Implementați algoritmul de sortare [Bubble Sort](https://en.wikipedia.org/wiki/Bubble_sort). **bubbleSort ()**

**Exercițiile de la 1 la 5** sunt **obligatorii**. Conceptele explorate sunt esențiale pentru obținerea notei **minime** de promovare.

**Vă recomandăm, pentru a crește șansele de a obține o notă cât mai mare să explorați și următoarele exerciții:**

1. (**selectionSort.c**) Implementați algoritmul de sortare [Selection Sort](https://en.wikipedia.org/wiki/Selection_sort). **selectionSort()**
2. (**bogoSort.c**) Implementați algoritmul de sortare [Bogosort](https://en.wikipedia.org/wiki/Bogosort). **bogoSort()**

**Exemplu afișare:**

**valoare0**

**valoare1**

**valoare2**

**valoare6**

**(null)**

**valoareNOUA**

**valoare2**

**(null)**

**Binary search position of 2 is 1**

**Binary search position of 1 is -1**

**0 0 0 1 1 3 3 4 4 6 7 11 11 13 13 13 15 15 15 16 16 16 16 19 20 21 21 24 25 25 26 28 28 28 31 31 32 34 35 35 35 37 38 38 39 40 41 41 42 43 43 43 47 47 48 53 53 54 54 54 57 59 62 63**

**0 0 0 1 1 3 3 4 4 6 7 11 11 13 13 13 15 15 15 16 16 16 16 19 20 21 21 24 25 25 26 28 28 28 31 31 32 34 35 35 35 37 38 38 39 40 41 41 42 43 43 43 47 47 48 53 53 54 54 54 57 59 62 63**

**Este sortat corect**