

Санкт-Петербургский политехнический университет Петра Великого  
Физико-механический институт  
Высшая школа фундаментальных физических исследований

**Лабораторная работа**

**Программирование алгоритмов поиска и сортировки на C++  
По дисциплине "Специальный практикум"**

Выполнил  
студент гр. 5040302/10301

С. А. Буланова

Научный руководитель:  
д.ф.-м.н.

Я. А. Бердников

«\_\_\_\_\_» \_\_\_\_\_ 2021 г.

Санкт-Петербург  
2021

## **Задание**

### **1. Вычисление чисел Фибоначчи**

Написать программу, выводящую первые  $n$  чисел Фибоначчи. Каждое число в последовательности Фибоначчи вычисляется как сумма двух предыдущих. Первые два члена последовательности равны 0 и 1.

### **2. Двоичный поиск**

Пусть задан отсортированный массив. Можно рассматривать числа массива как границы бинов гистограммы. Написать программу, которая возвращает номер бина, в который попадает введенное число.

### **3. Сортировка массива**

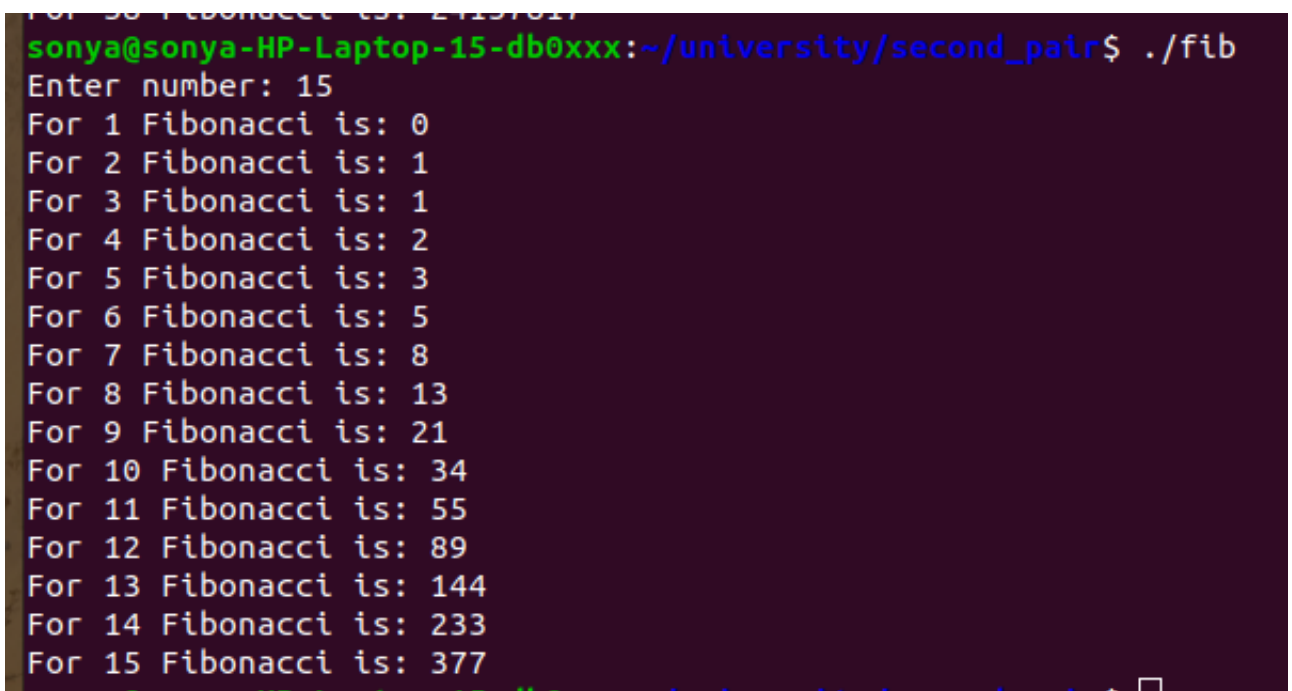
Написать программу, сортирующую массив любым методом.

## Задание 1

Приведем код, в котором числа Фибоначчи вычисляются через массив.

```
1 //fibonacci trough array
2 #include <iostream>
3 using namespace std;
4
5 int main(){
6     int n;
7     size_t Fib[10000];
8     cout << "Enter number: ";
9     cin >> n;
10    for (int i = 0; i < n; i++){
11        Fib[0] = 0;
12        Fib[1] = 1;
13        if (i > 1){
14            Fib[i] = Fib[i - 1] + Fib[i - 2];
15        }
16        cout << "For " << i + 1 << " " << "Fibonacci is: " << Fib[i] << '\n';
17    }
18 }
```

Вывод программы для первых 15 чисел Фибоначчи:



```
sonya@sonya-HP-Laptop-15-db0xxx:~/university/second_pair$ ./fib
Enter number: 15
For 1 Fibonacci is: 0
For 2 Fibonacci is: 1
For 3 Fibonacci is: 1
For 4 Fibonacci is: 2
For 5 Fibonacci is: 3
For 6 Fibonacci is: 5
For 7 Fibonacci is: 8
For 8 Fibonacci is: 13
For 9 Fibonacci is: 21
For 10 Fibonacci is: 34
For 11 Fibonacci is: 55
For 12 Fibonacci is: 89
For 13 Fibonacci is: 144
For 14 Fibonacci is: 233
For 15 Fibonacci is: 377
```

## Задание 2

Код двоичной сортировки:

```
1  #include <iostream>
2  #include <fstream>
3
4  using namespace std;
5  int BinarySearch(int arr[100], int idx, int search){
6      int l = 0;
7      int r = idx;
8      int m;
9      while (l <= r){
10         m = (l + r) / 2;
11         if (arr[m] == search){
12             return m;
13         }
14         else if (arr[m] > search) {
15             r = m - 1;
16         }
17         else if (arr[m] < search){
18             l = m + 1;
19         }
20     }
21     return m;
22 }
23
24 int main(){
25     ifstream in;
26     in.open("binning.txt");
27     int nLines = 0;
28     int bord, num;
29     int ary[100];
30
31     while (in >> bord){
32         ary[nLines] = bord;
33         //cout << ary[nLines] << '\n';
34         nLines++;
35     }
36     in.close();
37
```

```

38     cout << "Enter element: ";
39     cin >> num;
40     cout << BinarySearch(ary, nLines, num) << '\n';
41 }

```

Чтение массива осуществляется из файла binning.txt, пусть массив ary[] равен:

```
1 ary[] = {1, 7, 10, 12, 15, 21, 35, 39, 44, 56, 69, 72, 78, 89, 94, 97, 100};
```

Нумерация бинов начинается с единицы. Пример выполнения кода:

```

sonya@sonya-HP-Laptop-15-db0xxx:~/university/second_pair$ g++ -o search array.cpp
sonya@sonya-HP-Laptop-15-db0xxx:~/university/second_pair$ ./search
Enter element: 37
7
sonya@sonya-HP-Laptop-15-db0xxx:~/university/second_pair$ ./search
Enter element: 94
14

```

### Задание 3

Код, осуществляющий сортировку массива методом быстрой сортировки:

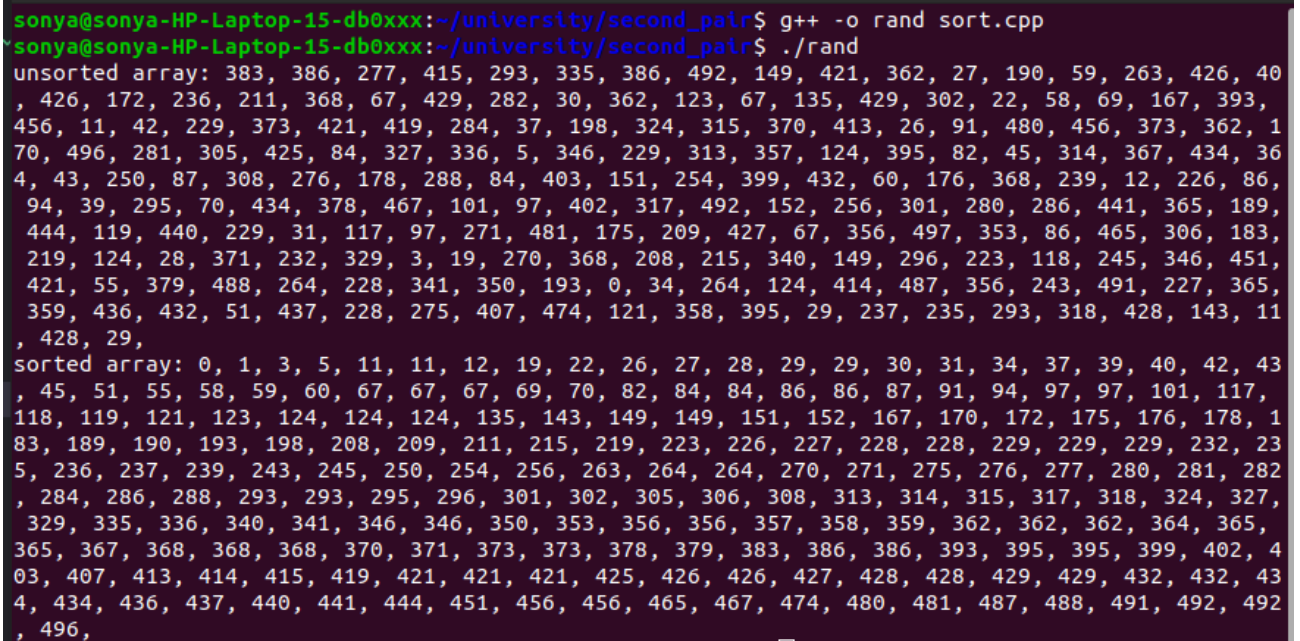
```
1  #include <iostream>
2  #include <fstream>
3  using namespace std;
4  #define ARR_SIZE 500
5
6  void SortArray(int *arr, int left, int right){
7      int i = left, j = right;
8      int cng, pivot = arr[(left+right)/2];
9      while (i <= j)
10     {
11         while (arr[i] < pivot) i++;
12         while (arr[j] > pivot) j--;
13
14         if (i <= j)
15         {
16             if (arr[i] > arr[j])
17             {
18                 cng = arr[i];
19                 arr[i] = arr[j];
20                 arr[j] = cng;
21             }
22             i++; j--;
23         }
24     }
25     if (left < j) SortArray (arr, left, j);
26     if (i < right) SortArray (arr, i, right);
27 }
28
29 int main(){
30     ifstream in;
31     in.open("unsorted.txt");
32     int nLines = 0;
33     int bord;
34     int arr[ARR_SIZE];
35
36     cout << "unsorted array: ";
37     while (in >> bord){
```

```

38     arr[nLines] = bord;
39     cout << arr[nLines] << ", ";
40     nLines++;
41 }
42 in.close();
43
44 cout << '\n';
45 cout << "sorted array: ";
46 SortArray(arr, 0, nLines);
47 for (int i = 0; i < nLines; i++){
48     cout << arr[i] << ", ";
49 }
50 cout << '\n';
51 }

```

Выведем неотсортированный и отсортированный массив:



```

sonya@sonya-HP-Laptop-15-db0xxx:~/university/second_pair$ g++ -o rand sort.cpp
sonya@sonya-HP-Laptop-15-db0xxx:~/university/second_pair$ ./rand
unsorted array: 383, 386, 277, 415, 293, 335, 386, 492, 149, 421, 362, 27, 190, 59, 263, 426, 40
, 426, 172, 236, 211, 368, 67, 429, 282, 30, 362, 123, 67, 135, 429, 302, 22, 58, 69, 167, 393,
456, 11, 42, 229, 373, 421, 419, 284, 37, 198, 324, 315, 370, 413, 26, 91, 480, 456, 373, 362, 1
70, 496, 281, 305, 425, 84, 327, 336, 5, 346, 229, 313, 357, 124, 395, 82, 45, 314, 367, 434, 36
4, 43, 250, 87, 308, 276, 178, 288, 84, 403, 151, 254, 399, 432, 60, 176, 368, 239, 12, 226, 86,
94, 39, 295, 70, 434, 378, 467, 101, 97, 402, 317, 492, 152, 256, 301, 280, 286, 441, 365, 189,
444, 119, 440, 229, 31, 117, 97, 271, 481, 175, 209, 427, 67, 356, 497, 353, 86, 465, 306, 183,
219, 124, 28, 371, 232, 329, 3, 19, 270, 368, 208, 215, 340, 149, 296, 223, 118, 245, 346, 451,
421, 55, 379, 488, 264, 228, 341, 350, 193, 0, 34, 264, 124, 414, 487, 356, 243, 491, 227, 365,
359, 436, 432, 51, 437, 228, 275, 407, 474, 121, 358, 395, 29, 237, 235, 293, 318, 428, 143, 11
, 428, 29,
sorted array: 0, 1, 3, 5, 11, 11, 12, 19, 22, 26, 27, 28, 29, 29, 30, 31, 34, 37, 39, 40, 42, 43
, 45, 51, 55, 58, 59, 60, 67, 67, 67, 69, 70, 82, 84, 84, 86, 86, 87, 91, 94, 97, 97, 101, 117,
118, 119, 121, 123, 124, 124, 124, 135, 143, 149, 149, 151, 152, 167, 170, 172, 175, 176, 178, 1
83, 189, 190, 193, 198, 208, 209, 211, 215, 219, 223, 226, 227, 228, 228, 229, 229, 229, 232, 23
5, 236, 237, 239, 243, 245, 250, 254, 256, 263, 264, 264, 270, 271, 275, 276, 277, 280, 281, 282
, 284, 286, 288, 293, 293, 295, 296, 301, 302, 305, 306, 308, 313, 314, 315, 317, 318, 324, 327,
329, 335, 336, 340, 341, 346, 346, 350, 353, 356, 356, 357, 358, 359, 362, 362, 362, 364, 365,
365, 367, 368, 368, 368, 370, 371, 373, 373, 378, 379, 383, 386, 386, 393, 395, 395, 399, 402, 4
03, 407, 413, 414, 415, 419, 421, 421, 421, 425, 426, 426, 427, 428, 428, 429, 429, 432, 432, 43
4, 434, 436, 437, 440, 441, 444, 451, 456, 456, 465, 467, 474, 480, 481, 487, 488, 491, 492, 492
, 496,

```