

Файл с исходным кодом с расширением «.сpp» добавьте в ответ на задание 6.2.

### ***Вариант 1***

Разработайте алгоритм инвертирования массива целых чисел, заданного генератором псевдослучайных чисел в диапазоне  $[0, 10]$ . Размер массива не известен заранее и указывается пользователем. Инвертирование – расположение элементов в обратном порядке, например, для набора чисел. Например: последовательность 1,2,3 переходит 3,2,1. Вывод массива в обратном порядке не допускается. Выделение дополнительной динамической памяти (дополнительный массив) при инвертировании не допускается. В реализации должно присутствовать две функции помимо главной. Первая для отображения массива. Вторая для инвертирования массива.

*Пример использования приложения*

```
>pr.exe 8
9 1 2 4 7 2 5 2
2 5 2 7 4 2 1 9
```

### ***Вариант 2***

Создать массив, который будет содержать все уникальные элементы произвольного массива, заданного генератором псевдослучайных чисел в диапазоне  $[0, k]$ . Размер массива и значение  $k$  указываются пользователем. Для поиска элементов в уникальном массиве используйте двоичный поиск. Сортировку использовать нельзя.

*Пример использования приложения*

```
>pr.exe n 11 k 10
1 7 7 2 8 9 1 10 2 2 6
1 2 6 7 8 9 10
```

### ***Вариант 3***

Напишите программу, которая для указанных пользователем чисел выводит следующие значения: число введенных значений, отсортированные по возрастанию значения, **среднее**, **медиану**, **среднеквадратичное отклонение**, минимальное и максимальное число.

*Пример использования приложения*

```
>pr.exe 1 7 3 2 8 9 1 10 2 2 6 1 4 5 4
n: 15
1 1 1 2 2 2 3 4 4 5 6 7 8 9 10
mean: 4.3333
median: 4.0
min: 1
max: 10

>pr.exe 4 2 1 3
```

```
n: 4
1 2 3 4
mean: 2.5
median: 2.5
min: 1
max: 4
```

#### *Вариант 4*

Напишите программу, которая принимает на вход  $n$  натуральных чисел, разделенных пробелами. Программа должна создать динамический массив и сохранить введенные целые числа в этот массив. Затем программа должна найти число с наибольшей суммой цифр и число с наименьшей суммой цифр в массиве и вывести их на экран.

*Пример использования приложения*

```
>pr.exe 11 22 123 456 12
4+5+6 = 15
1+1 = 2
```

#### *Вариант 5*

Напишите программу, которая принимает следующие аргументы командной строки:  $n$  – количество элементов массива, минимальное значение и максимальное значение для генератора случайных чисел. Программа должна создать динамический массив размером  $n$  и заполнить его случайными целыми числами, находящимися в указанном диапазоне. Затем программа должна вывести на экран созданный массив, заполненный генератором псевдослучайных чисел, и затем вывести все простые числа в массиве и факторизацию каждого не простого числа.

*Пример использования приложения*

```
>pr.exe 15 1 200
generated: 138 97 49 152 108 31 168 138 198 78 145 29 146 71 133
prime: 97, 31, 29, 71
138: 2 * 3 * 23
49: 7 * 7
152: 2 * 2 * 2 * 19
108: 2 * 2 * 3 * 3 * 3
168: 2 * 2 * 2 * 3 * 7
138: 2 * 3 * 23
198: 2 * 3 * 3 * 11
78: 2 * 3 * 13
145: 5 * 29
146: 2 * 73
133: 7 * 19
```

## Вариант 6

Напишите программу, которая принимает следующие аргументы командной строки:  $n$  – количество элементов последовательности, минимальное значение для генератора случайных чисел, максимальное значение для генерации случайных чисел. Программа должна создать динамический массив размером  $n$  и заполнить его случайными натуральными числами, находящимися в указанном диапазоне. Затем программа должна вывести на экран сгенерированную числовую последовательность и вывести на экран максимальную возрастающую подпоследовательность. Максимальная возрастающая подпоследовательность – это последовательность элементов, в которой каждый следующий элемент больше предыдущего, и сумма элементов максимальна среди всех возможных возрастающих подпоследовательностей.

*Пример использования приложения*

```
>pr.exe 16 1 17
0 8 4 12 2 10 6 14 1 9 5 13 3 11 7 15
longest increasing subsequence: 0 4 6 9 13 15
```

## Вариант 7

Напишите программу, которая вычисляет число  $\pi$  с заданной точностью с использованием метода Монте-Карло. Программа должна принимать один аргумент командной строки – точность, а выводить число итераций, на котором была получена данная точность.

*Пример использования приложения*

```
>pr.exe 0.0001
found 3.141676 after 79583 iterations
```

## Вариант 8

Напишите программу, которая для введенных пользователем символов английского алфавита формирует массив и выполняет сортировку элементов по возрастанию. Затем для введенного пользователем символа программа выполняет поиск индекса для вставки элемента в сформированный массив таким образом, чтобы не нарушался возрастающий порядок последовательности. Для поиска необходимо использовать двоичный поиск. Таким образом, если символ уже присутствует в массиве, он будет продублирован, а если отсутствует, то будет добавлен.

*Пример использования приложения*

```
>pr.exe d g a c j b d f c d d f g h j h a b

initial array
a b c c d d d f f g g h h j j
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14

enter symbol (or 0 to exit): e
found index 7
```

```
new array
a b c c d d d e f f g g h h j j
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

enter symbol (or 0 to exit): k
found index 16
new array
a b c c d d d e f f g g h h j j k
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

enter symbol (or 0 to exit): 0
bye!
```

### *Вариант 9*

Создать массив, который будет содержать все уникальные элементы произвольного массива, заданного генератором псевдослучайных чисел в диапазоне  $[0, k]$ . Размер массива и значение  $k$  указываются пользователем. Для поиска уникальных элементов воспользуйтесь одной из рассмотренных ранее сортировок.

#### *Пример использования приложения*

```
>pr.exe n 11 k 10
1 7 7 2 8 9 1 10 2 2 6
1 2 6 7 8 9 10
```