

# T2C1

## 1.3.1

Строка: `//100// * 1000 = 100100100100...`

Шаблон: 0001

Поиск будет неудачным, проверит всю строку.

Посчитаем сравнения для первой 100:

для 1: 1 сравнение

для 0: 3 сравнения

для 0: 2 сравнения

Повторится 1000 раз  $\Rightarrow (1 + 3 + 2) * 1000 = 6000$

## 1.3.2

```
1  B[n][n] - множество букв
2  Слово[m] - множество которое представляет собой слово
3  направления = [(1,0)(0,1)(-1,0)(0,-1)(1,1)(-1,-1)(1,-1)(-1,1)]
4  для i = 0 до n - 1
5      для j = 0 до n - 1
6          если B[i][j] == C[0], то
7              Для каждого (di, dj) в направления:
8                  совпадение = истина
9                      Для k от 1 до длина слова - 1
10                         ni = i + di*k
11                         nj = j + dj*k
12                         если ni < 0 или ni >= n или nj < 0 или nj >= n:
13                             совпадение = ложь
14                             прервать
15                         если B[ni][nj] != C[k], то
16                             совпадение = ложь
17                             прервать
18                     если совпадение = истина:
19                         слово найдено
```

## 1.5.1

```
1  x[n] - множество точек на оси координат
2  минимальное расстояние = |x[0] - x[1]|
3  для i от 1 до длина x[n] - 1
4      для j от 2 до длина x[n] - 2
5          если x[i] == x[j]
6              минимальное расстояние = 0
7              остановить
8      если | x[i] - x[j] | < минимальное расстояние, то
```

## 1.5.2

```

1  x[n] – множество точек на оси координат
2  макс_расст =  $|x[0] - x[1]|$ 
3  для i от 1 до длина x[n] – 1
4      для j от 2 до длина x[n] – 2
5          остановить
6      если  $|x[i] - x[j]| > \text{макс\_расст}$ , то
7          макс_расст =  $|x[i] - x[j]|$ 

```

## 1.5.3

```

1  Ввод N
2  мин_площадь_поверхности ← бесконечность
3
4  Для a от 1 до кубического корня из N:
5      Если  $N \bmod a \neq 0$ :
6          Продолжить
7
8      Для b от a до квадратного корня из  $(N / a)$ :
9          Если  $(N / a) \bmod b \neq 0$ :
10             Продолжить
11
12              $c \leftarrow N / (a * b)$ 
13             площадь ←  $2 * (a * b + b * c + a * c)$ 
14
15             Если площадь < мин_площадь_поверхности:
16                 мин_площадь_поверхности ← площадь
17
18  Вывод мин_площадь_поверхности

```

## 1.5.4

```

1  Ввод N, массив A[1..N]
2  общая_сумма ← сумма всех A[i]
3
4  Если общая_сумма mod 2  $\neq 0$ :
5      Вывод "Решение невозможно"
6      Завершить
7
8  целевая_сумма ← общая_сумма / 2
9
10 Для mask от 1 до  $(2^N - 2)$ : // исключаем пустое и полное множество
11     сумма ← 0
12     Для i от 1 до N:
13         Если i-й бит mask равен 1:

```

```

14         сумма ← сумма + A[i]
15
16     Если сумма = целевая_сумма:
17         Вывод "Подмножество 1:"
18         Для i от 1 до N:
19             Если i-й бит mask равен 1:
20                 Вывод A[i]
21
22         Вывод "Подмножество 2:"
23         Для i от 1 до N:
24             Если i-й бит mask равен 0:
25                 Вывод A[i]
26
27     Завершить
28
29     Вывод "Решение не найдено"

```

## 1.7.1

### Сортировка выбором (Selection Sort) — неустойчивая

- При выборе минимального элемента и его обмене с текущим может нарушиться порядок равных элементов.
- Пример: если в массиве два одинаковых значения с разными дополнительными метками (например, (5, A) и (5, B)), они могут поменяться местами — порядок нарушится. **Пузырьковая сортировка (Bubble Sort) — устойчивая**
- Обмен происходит только между соседними элементами.
- Равные элементы не переставляются, порядок сохраняется.

## 1.7.2

Реализация пузырьковой сортировки для связанных списков не является столь же эффективной, как для массивов.

В массивах доступ к элементам осуществляется по индексу за постоянное время  $O(1)$ , и обмен значений между элементами реализуется просто за счёт перестановки их значений.

В связанных списках доступ к элементам осуществляется путём последовательного обхода от начала списка, что требует времени  $O(n)$  в худшем случае. Кроме того, для обмена элементов в связанном списке необходимо изменять указатели, что делает операцию более сложной и затратной по времени, чем в массиве.

Таким образом, пузырьковая сортировка работает менее эффективно на связанных списках, чем на массивах.