

Вступительная работа — 2020

Информационно-технологический профиль

- Четыре задачи.
- За каждую можно получить 10 баллов.
- При оценивании могут учитываться все выкладки, записанные в чистовике и имеющие отношение к решению и пониманию сути задач.

1. Планирование очереди

Параллельная вычислительная система состоит из двух узлов: P и Q. Очередь запросов на обработку данных общая. Работа системы осуществляется по следующим правилам:

1. В каждый момент времени вычислительный узел может обрабатывать только один запрос. Запрос обрабатывается узлом непрерывно от начала до конца.

2. Вычислительные узлы работают непрерывно, принимая из очереди очередной запрос, как только завершают обработку предыдущего. Считается, что прием очередного запроса происходит мгновенно.

3. Освободившийся вычислительный узел всегда принимает следующий по порядку в очереди необработанный запрос, пока очередь не окажется пуста.

4. Если в один и тот же момент освободились оба вычислительных узла, то крайний в очереди запрос принимает вычислительный узел P, а следующий за ним (если очередь не стала пустой) – вычислительный узел Q. В таком же порядке вычислительные узлы принимают первые запросы в начале работы системы.

5. В очереди содержатся 10 запросов двух типов: 6 запросов типа A и 4 запроса типа B. В силу различной архитектуры вычислительные узлы обрабатывают различные запросы за различное время как указано в таблице:

| | Узел P | Узел Q |
|--|--------|--------|
| Время обработки запроса типа A, секунд | 3 | 11 |
| Время обработки запроса типа B, секунд | 7 | 8 |

Общим временем обработки запросов будем считать количество секунд, которое пройдет от момента приема и начала обработки вычислительными узлами первых запросов из очереди до того, как будет завершена обработка всех запросов.

Очевидно, что общее время, которое потребуется системе на обработку всех запросов, будет зависеть от их исходного расположения в очереди. Определите, за какое минимальное количество секунд могут быть обработаны все запросы при соблюдении указанных выше правил. В ответе укажите целое число и соответствующую очередь запросов.

2. Бэкапы

Резервное копирование — основной способ обеспечения сохранности данных. По выбору данных различают *Полное*, *Инкрементное* и *Дифференциальное* резервное копирование.

В некоторых файловых системах (NTFS, FAT) для определения, был ли изменен файл с момента резервного копирования, используется флаг – «архивный». Операционная система устанавливает флаг «архивный» при получении команды на создание или открытие файла. Этот флаг по-разному обрабатывается вышеперечисленными типами резервного копирования.

Полное – в резервное хранилище копируются все файлы, вне зависимости от того, установлен флаг «архивный» или нет, флаг сбрасывается для тех файлов, у которых он был установлен.

Инкрементное – резервируются только файлы с флагом «архивный» и флаг сбрасывается для этих файлов.

Дифференциальное – в резервное хранилище копируются только файлы с флагом «архивный», после копирования флаг **не** сбрасывается.

Системный администратор настроил следующее расписание резервного копирования:

| День | Время создания копии | Тип копии |
|------|----------------------|------------------|
| Пн | 01:00 | Полная |
| Вт | 01:00 | Дифференциальная |
| Ср | 01:00 | Инкрементная |
| Чт | 01:00 | Дифференциальная |
| Пт | 01:00 | Инкрементная |
| Сб | 01:00 | Дифференциальная |
| Вс | 01:00 | Инкрементная |

Восстановление данных из резервного хранилища осуществляется в два этапа:

1. Монтируется резервное хранилище. Время, необходимое на его монтирование зависит от суммарного объема данных, хранящихся в нем, и составляет **0,1** секунды на **1** мегабайт данных в резервном хранилище.

2. Данные из смонтированного хранилища копируются в место восстановления. Для получения только актуальных данных, копируется *полная* копия и все существующие на момент копирования *инкрементные* копии. Также копируется последняя *дифференциальная* копия, если на момент восстановления она является последней сделанной резервной копией.

Скорость копирования данных из смонтированного резервного хранилища – **15** МБайт/с. Перед началом создания полной резервной копии в понедельник резервное хранилище не содержало никаких данных, а объем данных в файловом хранилище составил **1484** МБайт. Какое время будет затрачено на восстановление данных из резервной копии в пятницу в 15:00 при условии, что каждый день изменялись или добавлялись разные файлы: в понедельник объем измененных или добавленных файлов составил **138** МБайт, во вторник – **196** МБайт, в среду – **457** МБайт, в четверг – **20** МБайт, в пятницу – **459** МБайт. В ответе укажите целое число секунд.

Примечание: данные изменяются строго в рабочее время с 9:00 до 18:00.

3. BF

Язык программирования «Ох, мой мозг» напоминает один из известнейших эзотерических языков программирования и содержит следующие операторы:

| | |
|---|--|
| > | перейти к следующей ячейке |
| < | перейти к предыдущей ячейке |
| + | увеличить значение в текущей ячейке на 1 |
| - | уменьшить значение в текущей ячейке на 1 |
| [| если значение текущей ячейки ноль, перейти вперёд по тексту программы на ячейку, следующую за соответствующей] (с учётом вложенности) |
|] | если значение текущей ячейки не ноль, перейти назад по тексту программы на символ [(с учётом вложенности) |
| . | Напечатать значение из текущей ячейки |

Несмотря на свой минимализм, язык обладает тьюринг-полнотой — то есть на нем определенно можно реализовать любую вычислимую функцию.

Одна из программ, записанная на этом языке попала вам в работу. В качестве ответа на задание требуется написать число, которое напечатает следующая программа:

```
[ -> [ -> ++< ] > [ -> +++< ] > [ -> ++++< ] > [ -> +++++< ] <<<< ] >>>> .
```

Исполнение программы начинается с первой ячейки, в момент запуска программы ячейки памяти содержали следующие значения:

| | | | | | | | | |
|--------------|---|-----|---|---|---|---|---|---|
| Номер ячейки | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| Значение | 2 | 158 | 0 | 0 | 0 | 0 | 0 | 0 |

Примечание: под вложенностью операторов [и] подразумевается однозначное соответствие оператора [к оператору] и наоборот. Например, для программы [->+ [-] <] первая открывающая квадратная скобка соответствует второй закрывающей квадратной скобке, а вторая открывающая квадратная скобка соответствует первой закрывающей квадратной скобке.

4. Консольный график

Распознавание образов – штука непростая. Для того, чтобы определить, что изображено на рисунке, нужна предварительная обработка. Иннокентий решил начать изучение с анализа ASCII-графиков, представленных в виде прямоугольной таблиц $n \times m$. Таблицы состоят из символов «*» и «.». Началом координат считается левая верхняя клетка — точка $(0, 0)$, осью OX (направлена **слева направо**) — верхняя строка таблицы, а осью OY (направлена **сверху вниз**) — левый столбец.

Клетка (x, y) таблицы отвечает за точку (x, y) на графике функции, и если в этой клетке стоит «*», то $f(x) = y$, в противном случае в клетке стоит «.». Гарантируется, что функция, график которой дан Вам, непрерывна и однозначно определена на всем промежутке, то есть:

- в каждом столбце таблицы стоит ровно один символ «*»;
- в соседних столбцах символы «*» находятся либо в соседних по стороне, либо в соседних по углу клетках.

Для начала, Иннокентий хочет найти количество локальных максимумов в нем, то есть таких x , что $f(x - 1) < f(x) > f(x + 1)$. Если одно из значений $f(x - 1)$ или $f(x + 1)$ не определено, считается, что эта часть двойного неравенства выполняется.

Формат ввода

На вход программы поступает два натуральных числа n и m — количество строк и количество столбцов в таблице соответственно ($1 \leq n, m \leq 100$). В каждой из следующих n строк содержится строка из m символов — описание таблицы. Гарантируется, что таблица представляет собой график функции, описанной в условии.

Формат вывода

Программа должна выводить одно число — количество локальных максимумов в данном графике функции.

Пример 1

| Ввод | Вывод |
|--|-------|
| 3 7 * . * . . * . * . * . * . . . * . . | 2 |

Пример 2

| Ввод | Вывод |
|--|-------|
| 3 5 * * * * * | 1 |

Примечания:

В первом примере два локальных максимума ($x = 1$ и $x = 4$):

$$f(0)=0 < f(1)=1 > f(2)=0;$$
$$f(3)=1 < f(4)=2 > f(5)=1.$$

Во втором примере один локальный максимум ($x = 4$):

$$f(3)=1 < f(4)=2, \text{ а } f(5) \text{ не определено.}$$