|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ *Робототехники и комплексной автоматизации*

КАФЕДРА *Системы автоматизированного проектирования (РК-6)*

**ОТЧЕТ О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ**

по дисциплине: «Информатика»

|  |  |  |
| --- | --- | --- |
| Студент |  | Камалов Антон Павлович |
| Группа |  | РК6-15Б |
| Тип задания |  | Лабораторная работа №4 |
| Тема лабораторной работы |  | Работа со строками, Вариант 8D |

Студент **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_ \_**

*подпись, дата фамилия, и.о.*

Преподаватель **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_ \_\_**

*подпись, дата фамилия, и.о.*

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Москва, 2022 г.*

# Вариант задания

Разработать программу перечисления всех месяцев любого указанного или текущего года, в которых заданное число выпадает на одинаковый день недели. Число и номер года должны передаваться программе через аргументы командной строки её вызова. При этом если год не указан, то должен рассматриваться текущий год. Результат поиска должен отображаться строкой потока стандартного вывода, где названия всех месяцев представлены латинской аббревиатурой.

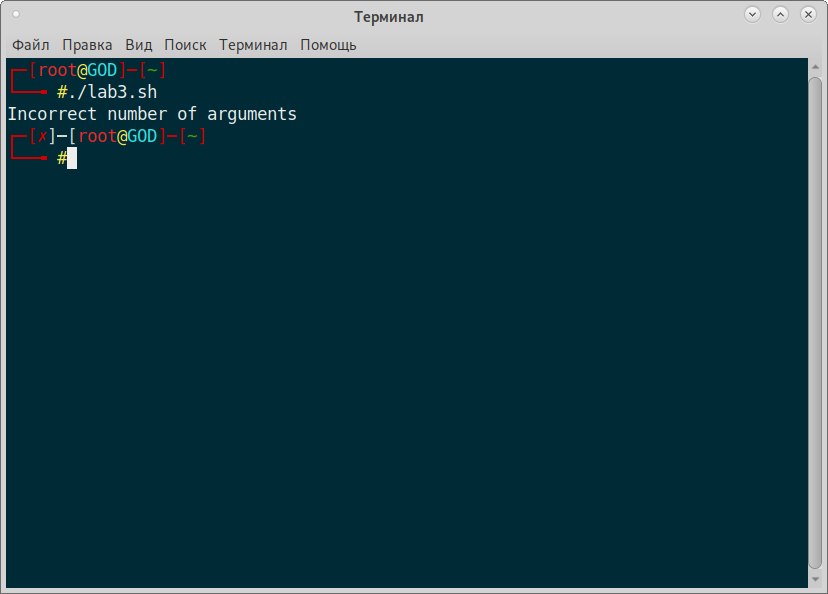
# Описание решения

1. В командную строку подаётся 2 аргумента – число и номер года. Если номер года отсутствует, то в качестве второго аргумента подаётся номер текущего года.
2. Производится проверка на число аргументов, правильность введённого года (если он представлен как аргумент), правильность числа. Также проверяется, что номер года больше нуля.
3. Создаётся массив DAYS для добавления подходящих результатов, также массив month с порядковым номером каждого месяца.
4. Производится поиск двух дат, чьи числа выпадают на одинаковый день недели. Если одна из дат оказывается в неверном формате, то итерация прекращается и происходит переход к следующему (если он существует) месяцу.
5. Производится вывод дней недели и месяцев, чьи результаты удовлетворяют условию поиска.

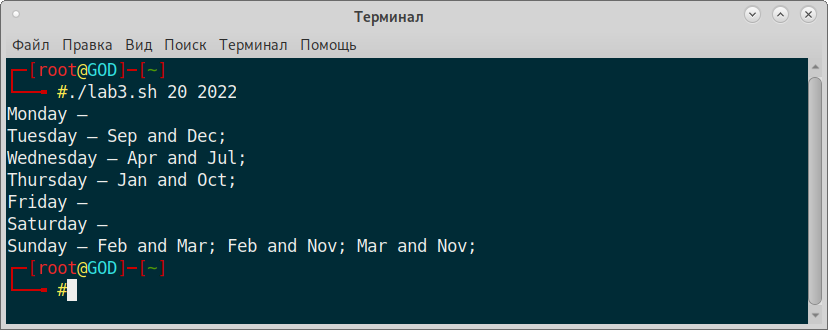
# Текст программы

|  |  |
| --- | --- |
| 01:  02:  03:  04:  05:  06:  07:  08:  09:  10:  11:  12:  13:  14:  15:  16:  17:  18:  19:  20:  21:  22:  23:  24:  25:  26:  27:  28:  29:  30:  31:  32:  33:  34:  35:  36:  37:  38:  39:  40:  41:  42:  43:  44:  45:  46:  47:  48:  48:  49:  50:  51:  52:  53:  54:  55:  56:  57:  58:  59:  60:  61:  62:  63:  64:  65:  66:  67:  68:  69:  70:  71:  72:  73:  74:  75:  76:  77:  78: | #!/bin/bash  # Установка языка терминала  LANG=en\_us\_8859\_1  # Получение текущего года  NOW=`date "+%Y"`  # Проверка на количество аргументов  case $# in      1)  DY=$1/$NOW;;      2)  expr $2 + 1 &>/dev/null          if [ $? -ne 0 ]; then              echo "$2 is not correct"              exit -1          fi          DY=$1/$2;;      \*) echo "Incorrect number of arguments"          exit -2  esac  # Проверка на правильность числа  expr $1 + 0 &>/dev/null  if [ $? -ne 0 -o $1 -gt 31 -o $1 -lt 1 ]; then      echo "$1 is not correct"      exit -3  fi  # Проверка на правильность года (он должен быть ≥ 0)  if [ $2 -lt 0 ]; then      echo "$2 is not correct"      exit -5  fi  DAYS=("Monday – " "Tuesday – " "Wednesday – " "Thursday – " "Friday – " "Saturday – " "Sunday – ")  month=(01 02 03 04 05 06 07 08 09 10 11 12)  # Индекс следующего элемента массива month  c=1  # M1 – текущий месяц из массива month  for M1 in ${month[@]}  do      # M2 – следующие после M1 месяцы      for M2 in ${month[@]:$c}      do          # Проверка на существование дат          date -d "$M1/$DY" &> /dev/null          if [ $? -ne 0 ]; then              continue          fi          date -d "$M2/$DY" &> /dev/null          if [ $? -ne 0 ]; then              continue          fi          # Дни недели двух дней разных месяцев          FIRST=`date -d "$M1/$DY" +%a`          SECOND=`date -d "$M2/$DY" +%a`          # Проверка на равенство дней недели          if [ $FIRST = $SECOND ]; then              ABR\_1=`date -d "$M1/$DY" +%b`              ABR\_2=`date -d "$M2/$DY" +%b`              INDEX=`expr $(date -d "$M1/$DY" +%u) - 1`              DAYS[INDEX]="${DAYS[INDEX]}$ABR\_1 and $ABR\_2; "          fi      done      c=`expr $c + 1`  done  # Вывод дней недели с соответствующими месяцами  for j in "${DAYS[@]}"  do      echo $j  done |

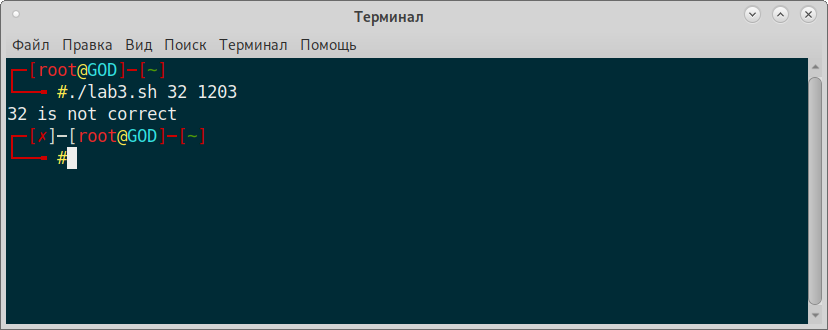
# Результаты работы программы



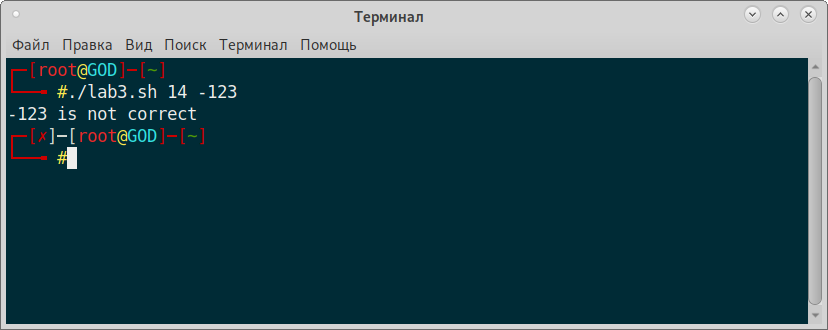
*Рисунок 1. Выполнение программы без аргументов*



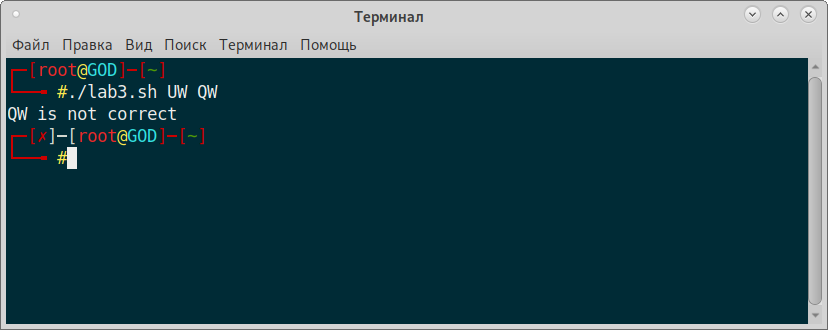
*Рисунок 2. Выполнение программы с корректными аргументами*



*Рисунок 3. Выполнение программы с некорректным числом*



*Рисунок 4. Выполнение программы с некорректным годом*



*Рисунок 5. Выполнение программы с некорректными аргументами*