

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук, Департамент программной инженерии
Дисциплина: «Архитектура вычислительных систем»

**ПРОГРАММА ВЫЧИСЛЕНИЯ
ДАТЫ ПРАВОСЛАВНОЙ ПАСХАЛИИ**

Пояснительная записка

Выполнила:
Абу Аль Лабан Надя,
студент гр. БПИ198.

Москва
2020

Содержание

1. Текст задания	2
2. Применяемые расчетные методы	3
2.1. Теория решения задания	3
2.2. Дополнительный функционал программы	3
3. Тестирование программы	4
3.1. Корректные значения	4
3.2. Некорректные значения	5
ПРИЛОЖЕНИЕ 1. Список литературы	6
ПРИЛОЖЕНИЕ 2. Код программы	7

1. Текст задания

Разработать программу вычисления даты православной Пасхалии для заданного года

2. Применяемые расчетные методы

2.1. Теория решения задания

Вычисление даты православной пасхалии производилось по математическому алгоритму Гаусса для вычисления даты Пасхи [1]. Согласно этому алгоритму, для определения даты Православной пасхи необходимо:

1. Разделить номер года на 19 и определить остаток от деления a .
2. Разделить номер года на 4 и определить остаток от деления b .
3. Разделить номер года на 7 и определить остаток от деления c .
4. Разделить сумму $19a + 15$ на 30 и определить остаток d .
5. Разделить сумму $2b + 4c + 6d + 6$ на 7 и определить остаток e .
6. Определить сумму $f = d + e$.
7. Если $f \leq 26$, то Пасха будет праздноваться $4 + f$ апреля; если $f > 26$, то Пасха будет праздноваться $f - 26$ мая.

Для простоты вычислений в программе вычисляется три значения:

- `coefA = (19 * (Y mod 19) + 15) mod 30` – остаток d из алгоритма.
- `coefB = 2 * (Y mod 4)` – часть остатка e из алгоритма.
- `coefB = 4 * (Y mod 7)` – вторая часть остатка e из алгоритма.

Затем мы работаем с регистром `ax`:

- Кладем в него `coefA` и умножаем на 6.
- Прибавляем `coefB`, `coefC` и 6 и берем остаток от деления на 7.

Теперь в `ax` лежит значение остатка e из алгоритма.

- Прибавляем `coefA`.

Теперь в `ax` лежит значение остатка f из алгоритма.

Далее все происходит в соответствии алгоритму. Регистр `ax` сравнивается с 26 и в зависимости от результата переходит к нужной метке (`april` или `may`) для подсчета и вывода ответа.

2.2. Дополнительный функционал программы

Пользователь вводит год, для которого нужно рассчитать дату Пасхалии. Поскольку дата рассчитывается выводится по новому стилю календаря, входные данные ограничиваются началом XX века.

При попытке ввести год до 1900 или строку программа сообщает пользователю о некорректном вводе и завершает работу.

3. Тестирование программы

При запуске программы выводится просьба ввести значение с указанием ограничения входных данных (рис. 1).

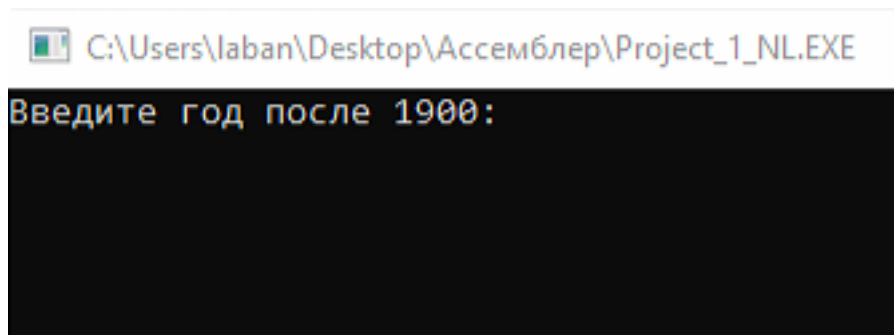


Рисунок 1. Запуск

3.1. Корректные значения

Для начала введем корректные значения и сверимся с православным календарем [2] для проверки правильности работы алгоритма. В календаре дата Пасхи отмечена красным цветом.

Начнем с граничного значения и введем 1900 (рис. 2). Дата совпадает с датой в календаре (рис. 3).

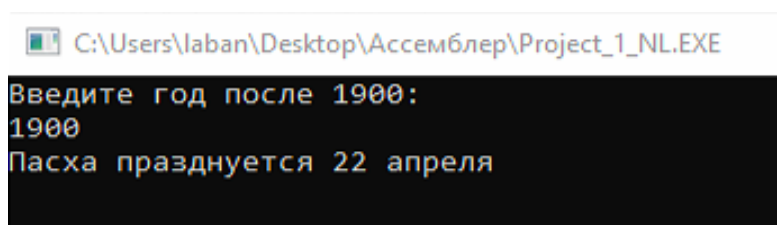


Рисунок 2. Вывод при входных данных 1900

<< Апрель 1900 >>

Пн	Вт	Ср	Чт	Пт	Сб	Вс
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30						

Рисунок 3. Православный календарь на апрель 1900

Теперь введем текущий год (рис. 4). Дата верная (рис. 5).

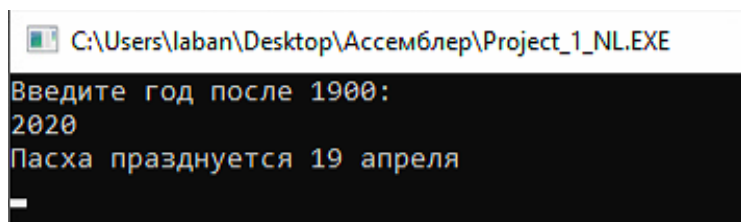


Рисунок 4. Вывод при входных данных 2020

<< Апрель 2020 >>

Пн	Вт	Ср	Чт	Пт	Сб	Вс
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

Рисунок 5. Православный календарь на апрель 2020

Введем год, который еще не наступил (рис. 6). Дата верна (рис. 7).

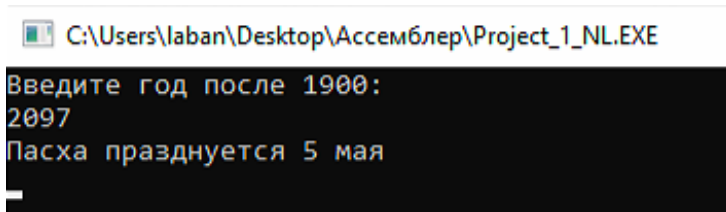


Рисунок 6. Вывод при входных данных 2020

<< Май 2097 >>

Пн	Вт	Ср	Чт	Пт	Сб	Вс
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

Рисунок 7. Православный календарь на апрель 2020

Таким образом, мы видим, что программа работает корректно на граничном значении, для лет, когда Пасхалия выпадает на апрель, и для лет, когда Пасхалия выпадает на май.

3.2. Некорректные значения

Чтобы проверить работу программы при некорректном вводе, нужно понять, какие случаи программа должна обрабатывать.

Для начала введем строковое значение, поскольку (рис. 8).

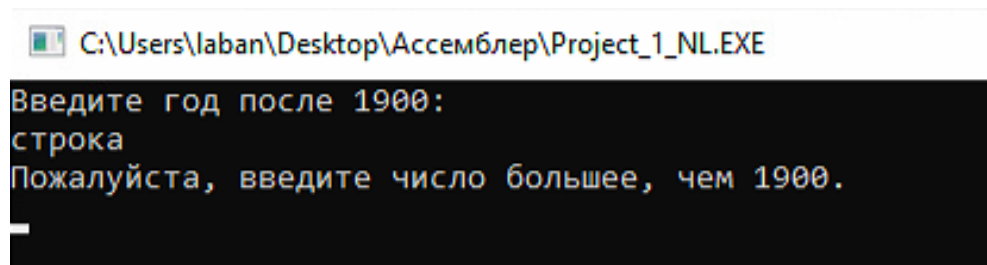


Рисунок 8. Ввод строкового значения

Теперь введем значение меньшее, чем ограничение (рис. 9).

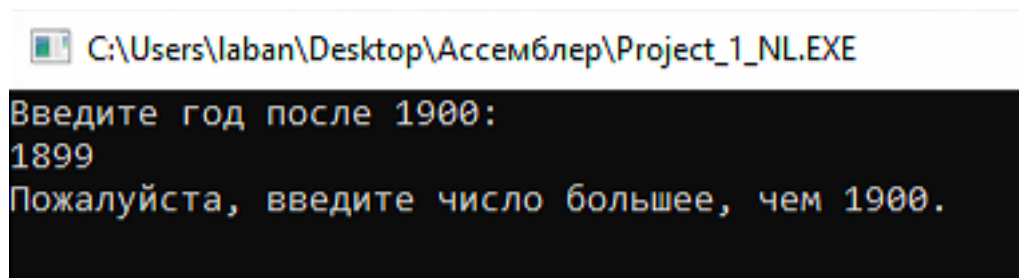


Рисунок 9. Ввод числа, не входящего в допустимые значения

ПРИЛОЖЕНИЕ 1

Список литературы

1. Алгоритм Гаусса вычисления даты пасхи. [Электронный ресурс] // Режим доступа: свободный, URL: https://ru.wikipedia.org/wiki/Алгоритм_Гаусса_вычисления_даты_Пасхи (дата обращения: 27.10.2020)
2. Православный календарь. [Электронный ресурс] // Режим доступа: свободный, URL: http://calendar.lenacom.spb.ru/orthodox_calendar (дата обращения: 27.10.2020)
3. Руководство по синтаксису FASM [Электронный ресурс]. // Режим доступа: свободный, URL: <http://flatassembler.narod.ru/fasm.htm> ^[L]_[SEP] (дата обращения: 27.10.2020)

ПРИЛОЖЕНИЕ 2

Код программы

```
; Разработать программу вычисления даты
; православной Пасхалии для заданного года
; Абу Аль Лабан Н. БПИ 198

format PE console          ; Формат выходного файла - консоль
entry start                ; Точка входа

include 'win32ax.inc'      ; Подключаемые модули

section '.data' data readable writable      ; Секция данных
    ru db 'Russian', 0

    strYear      db 'Введите год после 1900: ', 13, 10, 0
    strIncorrect db 'Пожалуйста, введите число большее, чем 1900.',13, 10, 0
    strApril     db 'Пасха празднуется %d апреля', 13, 10, 0
; Тексты для вывода в консоль
    strMay       db 'Пасха празднуется %d мая'', 13, 10, 0

    string db '%d', 0          ; Формат ввода

    year      dw ?              ; Введенный в консоль год

    coefA     dw ?
    coefB     dw ?              ; Коэффициенты для расчета даты
    coefC     dw ?

    result     db 256 dup(?)
    path       db 256 dup("?")
    opsymbol   dw ?
```



```

; -----
section '.code' code readable executable      ; Секция кода
start:

    cinvoke setlocale, 0, ru                  ; Устанавливаем русскую локаль

    push strYear
    call [printf]                             ; Запрашиваем ввод года

    push year
    push string                               ; Считываем год
    call [scanf]

    call pashalia

; -----

proc pashalia
    mov ax, word[year]

    cmp ax, 1900
    jl incorrect                             ; Проверяем корректность ввода

    mov bl, 19
    div bl
    mov al, ah
    mov bl, 19
    mul bl

    ; Считаем коэффициент A
    add ax, 15                               ;  $A = (19 * (Y \bmod 19) + 15) \bmod 30$ 
    mov bl, 30
    div bl
    mov al, ah
    cbw

    mov word[coefA], ax

```

```

mov ax, word[year]
mov dx, 0
mov bx, 4
div bx                                ; Считаем коэффициент В
                                        ;  $B = 2 * (Y \bmod 4)$ 

mov ax, dx
mov bx, 2
mul bx

mov word[coefB], ax

mov ax, word[year]
mov dx, 0
mov bx, 7
div bx                                ; Считаем коэффициент С
                                        ;  $C = 4 * (Y \bmod 7)$ 

mov ax, dx
mov bx, 4
mul bx
mov word[coefC], ax

mov ax, word[coefA]
mov bl, 6                            ; Кладем А в регистр и умножаем на 6
mul bl

add ax, 6
add ax, word[coefB]                  ; Прибавляем 6, В и С
add ax, word[coefC]

mov bl, 7
div bl                                ; Берем остаток от деления на 7

mov al, ah

```

```

    cbw

    add ax, word[coefA]          ; Прибавляем A
    cwde                        ; Получается A + ((B + C + 6A + 6) mod

7)

    cmp eax, 26
    jg may                      ; В зависимости от значения переходим
    jmp april                   ; В нужную метку

; -----

    april:
    add eax, 4                  ; Прибавляем 4 и получаем дату в
апреле
    push eax
    push strApril               ; Выводим ответ
    call [printf]
    add esp,8

    jmp finish

; -----

    may:
    sub eax, 26                 ; Отнимаем 26 и получаем дату в мае
    push eax
    push strMay                 ; Выводим ответ
    call [printf]
    add esp,8

    jmp finish

; -----

    incorrect:
    push strIncorrect
    call [printf]

```

```

        add esp,4                                ; Выводим сообщение об ошибке

        push 0
        call [getch]
        call [ExitProcess]

; -----

        finish:

        push 0                                ; Завершение программы
        call [getch]

        call [ExitProcess]

        ret
    endp

; -----

section '.idata' data readable import            ; Секция импортируемых данных
    library kernel, 'kernel32.dll', \
        msvcrt, 'msvcrt.dll'
    import kernel,\
        ExitProcess, 'ExitProcess'
    import msvcrt,\
        setlocale, 'setlocale',\
        printf, 'printf',\
        getch, '_getch',\
        scanf, 'scanf'

```