

**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук
Департамент программной инженерии

Микропроект

**ПРОГРАММА ВЫЧИСЛЕНИЯ ДАТЫ ПРАВОСЛАВНОЙ ПАСХАЛИИ ДЛЯ
ЗАДАННОГО ГОДА**

Пояснительная записка

Исполнитель
студент группы БПИ 199
М.А. Кузнецов

Москва 2020

СОДЕРЖАНИЕ

1. ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ	3
1.1 Постановка задачи	3
2. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ	3
2.1 Описание алгоритма	3
2.2 Обоснование формата выходных данных	3
2.3 Обоснование ограничений входных данных.....	3
2.4 Функционирование программы	3
3. ТЕСТЫ	4
3.1 Проверка нижней границы	4
3.2 Проверка верхней границы.....	4
3.3 Ввод корректного года.....	4
4. ИСПОЛЬЗОВАННАЯ ЛИТЕРАТУРА	5
ПРИЛОЖЕНИЕ	6

1. ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ

1.1 Постановка задачи

Программа должна вычислять дату православной Пасхалии для заданного года

2. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

2.1 Описание алгоритма

Для решения поставленной задачи был использован алгоритм Гаусса вычисления даты православной Пасхи [1]. Данный алгоритм включает в себя несколько шагов вычисления определённых чисел, а именно:

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 $f = d + e$.

В результате применения алгоритма получается число f . Если $f \leq 26$, то Пасха будет праздноваться $4 + f$ апреля; если $f > 26$, то Пасха будет праздноваться $f - 26$ мая (даты григорианского календаря).

2.2 Обоснование формата выходных данных

В качестве результата работы программа отображает дату православной Пасхи григорианского календаря. Выбран именно этот календарь, так как на данный момент он является основным в России и для пользователя будет удобнее получать дату Пасхи в григорианском календаре.

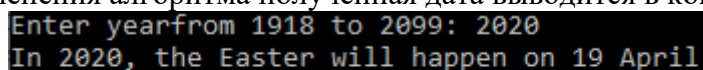
2.3 Обоснование ограничений входных данных

Программа получает на вход число от 1918 до 2099. Нижняя граница была выбрана именно такой, так как с 1918 года Россия перешла с юлианского на григорианский календарь, а в качестве результата работы программа отображает дату григорианского календаря.

В результате применения алгоритма Гаусса изначально получается дата юлианского календаря. Для получения даты в новом стиле необходимо прибавить 13 к дате в старом стиле. Но в будущем эта разница будет увеличиваться из-за разного количества високосных годов. Например, в 2100 – 2199 годах разница будет составлять уже 14 дней, а в 2300-2499 годах 16 дней и при этом интервал, когда у лет в интервале одна и та же разница с юлианским календарем, увеличивается. Из-за этого верхняя граница была выбрана 2099, чтобы разница с юлианским составляла 13 дней

2.4 Функционирование программы

Для получения даты православной Пасхалии необходимо ввести корректный год (с 1918 по 2099). После применения алгоритма полученная дата выводится в консоль (см. рис. 1)



```
Enter year from 1918 to 2099: 2020
In 2020, the Easter will happen on 19 April
```

Рисунок 1 – Функционирование программы

3. ТЕСТЫ

3.1 Проверка нижней границы

```
Enter yearfrom 1918 to 2099: -100  
Incorrect year = -100
```

Рисунок 2– Ввод отрицательных значений

```
Enter yearfrom 1918 to 2099: 1917  
Incorrect year = 1917
```

Рисунок 3– Ввод числа меньше 1918

```
Enter yearfrom 1918 to 2099: 1918  
In 1918, the Easter will happen on 5 May
```

Рисунок 4– Ввод нижней границы

3.2 Проверка верхней границы

```
Enter yearfrom 1918 to 2099: 2100  
Incorrect year = 2100
```

Рисунок 5– Ввод числа больше 2099

```
Enter yearfrom 1918 to 2099: 2099  
In 2099, the Easter will happen on 12 April
```

Рисунок 6– Ввод верхней границы

3.3 Ввод корректного года

```
Enter yearfrom 1918 to 2099: 2030  
In 2030, the Easter will happen on 28 April
```

Рисунок 7– Ввод 2030. Получение даты в апреле

```
Enter yearfrom 1918 to 2099: 1956  
In 1956, the Easter will happen on 6 May
```

Рисунок 8– Ввод 1956. Получение даты в мае

4. ИСПОЛЬЗОВАННАЯ ЛИТЕРАТУРА

- 1) Описание алгоритма Гаусса вычисления даты Пасхи [Электронный ресурс]. Режим доступа: https://ru.wikipedia.org/wiki/Алгоритм_Гаусса_вычисления_даты_Пасхи, свободный. (дата обращения: 28.10.20).
- 2) Курс «Инструкции процессоров Intel» / Команда DIV [Электронный ресурс]. Режим доступа: <http://www.av-assembler.ru/instructions/div.php>, свободный. (дата обращения: 28.10.20).
- 3) Материалы по дисциплине «Системное программирование» Криворожского колледжа НАУ / Команды безусловного и условного переходов в языке Ассемблер. [Электронный ресурс]. Режим доступа: <https://www.sites.google.com/site/sistprogr/lekcii1/lek9> свободный. (дата обращения: 28.10.20).
- 4) Практические примеры для программирования / Деление и умножение в Assembler. [Электронный ресурс]. Режим доступа: <https://codetown.ru/assembler/delenie-umnozhenie/> свободный. (дата обращения: 28.10.20).

ПРИЛОЖЕНИЕ

КОД ПРОГРАММЫ

```
format PE console
entry start

include 'win32a.inc'

; Кузнецов Михаил Александрович БПИ199
; 13 вариант
; Вычисление даты пасхи по введенному году
;-----
section '.data' data readable writable

    strYear    db 'Enter yearfrom 1918 to 2099: ', 0
    strIncorSize db 'Incorrect year = %d', 10, 0
    strPashaApril db 'In %d, the Easter will happen on %d April', 10, 0
    strPashaMay db 'In %d, the Easter will happen on %d May', 10, 0
    strScanInt   db '%d', 0

    f           dd 0
    a           dd 0
    b           dd 0
    c           dd 0
    d           dd 0
    e           dd 0
    year        dd 0

;-----
section '.code' code readable executable
start:
    ;Вводим год
    call YearInput
    ;Поиск числа a
    call count_a
    ;Поиск числа b
    call count_b
    ;Поиск числа c
    call count_c
    ;Поиск числа d
    call count_d
    ;Поиск числа e
    call count_e
    ;Поиск числа f
    call count_f

    ;Вывод вычисленной даты пасхи
    call printDate
finish:
    ; Завершениеработы программы
    call [getch]
    push 0
    call [ExitProcess]

;-----Ввод года-----
YearInput:
    push strYear
    call [printf]
    add esp, 4
```

```

    push year
    push strScanInt          ; Считывание года
    call [scanf]
    add esp, 8

    cmp [year], 1918
    jl failSize

    cmp [year], 2099
    jg failSize

    ret
failSize:                    ; Вывод сообщения о некорректной длине
    push [year]
    push strIncorSize
    call [printf]
    jmp finish
;-----Деление чисел и поиск их остатка-----
divide:
    mov eax, [esp+8]         ; Делимое
    cdq
    mov ebx, [esp+4]         ; Делитель
    idiv ebx

    ret
;-----Подсчет числа a-----
count_a:
    push [year]
    push 19
    call divide              ; Поиск остатка от деления года на 19
    mov [a], edx
    add esp, 8

    ret
;-----Подсчет числа b-----
count_b:
    push [year]
    push 4
    call divide              ; Поиск остатка от деления года на 4
    mov [b], edx
    add esp, 8

    ret
;-----Подсчет числа c-----
count_c:
    push [year]
    push 7
    call divide              ; Поиск остатка от деления года на 7
    mov [c], edx
    add esp, 8

    ret
;-----Подсчет числа d-----
count_d:
    mov ebx, [a]
    mov eax, 19
    mul bx                   ; Перемножение a и 19

```

```

    add eax, 15                ; Добавление к произведению 15
    push eax
    push 30
    call divide                ; Поиск остатка от деления a*19 + 15 на 30
    add esp, 8
    mov [d], edx

    ret

;-----Подсчет числа e-----
count_e:
    xor ecx, ecx

    mov ebx, [b]
    mov eax, 2
    mul ebx                    ; Перемножение b и 2
    add ecx, eax               ; Добавление к сумме b*2

    mov ebx, [c]
    mov eax, 4
    mul ebx                    ; Перемножение c и 4
    add ecx, eax               ; Добавление к сумме c*4

    mov ebx, [d]
    mov eax, 6
    mul ebx                    ; Перемножение d и 6
    add ecx, eax               ; Добавление к сумме d*6
    add ecx, 6                 ; Добавление к сумме 6

    push ecx
    push 7
    call divide                ; Поиск остатка от деления 2*b + 4*c + 6*d +
6 на 7
    mov [e], edx
    add esp, 8

    ret

;-----Подсчет числа f-----
count_f:
    mov eax, [d]
    mov ecx, [e]
    add eax, ecx               ; Сложение e и f
    mov [f], eax
    ret

;-----Вывод даты пасхи-----
printDate:
    mov eax, [f]
    cmp eax, 26                ; Если f > 26, то перейти на вывод даты в
марте. Иначе вывод даты в апреле
    ja printMay

    add eax, 4
    push eax
    push [year]
    push strPashaApril
    call [printf]              ; Вывод даты в апреле
    jmp outPrint               ; Переход к завершению вывода

printMay:

```



```

        sub eax, 26
        push eax
        push [year]
        push strPashaMay
        call [printf]                ; Вывод даты в марте

outPrint:                                ; Завершение вывода
        add esp, 12
        ret
;-----

```

```

section '.idata' import data readable
    library kernel, 'kernel32.dll',\
        msvcrt, 'msvcrt.dll'

```

```

include 'api\kernel32.inc'
import kernel,\
    ExitProcess, 'ExitProcess'
include 'api\kernel32.inc'
import msvcrt,\
    printf, 'printf',\
    scanf, 'scanf',\
    getch, '_getch'

```