

Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Алтайский государственный технический университет
им. И.И. Ползунова»

Факультет (институт) Информационных технологий

Кафедра Прикладная математика

Отчет защищен с оценкой _____

Е.Г. Боровцов

(подпись преподавателя) (инициалы, фамилия)

“ ” _____ 2022 г.

Отчет

по лабораторной (практической) работе № 3

«Подпрограммы и программные прерывания»

(название лабораторной (практической) работы)

по дисциплине Машино-зависимые языки программирования
(наименование дисциплины)

Студент группы ПИ-02 Чередов Р.А.

Преподаватель Е.Г. Боровцов

Барнаул 2022

Задание

Реализовать алгоритм с обязательным использованием подпрограмм.

Передачу параметров реализовать исключительно через стек, для обращения к параметрам в стеке

использовать имена параметров, присвоенные через EQU. Адресация параметров относительно BP,

при этом обязательно использование стандартной входной и выходной последовательности.

Память под локальные переменные процедуры выделить в окне стека.

Реализовать обращение к ним

относительно регистра BP с использованием имен, назначенных через директивы EQU.

В качестве индивидуального задания использовать задания из курса "Архитектура ВС", тема "подпрограммы"

Выражение:

21	$Y := P(Y, X^2) - P(Y, X-1) + P(Y, X)$, где $P(A, Z) = az^n - a^n + 1$;
----	---

Текст программы

```
.model tiny          ; MASM's support for COM files
```

```
.dosseg              ; Make sure you are using dos segment CODE, DATA + STACK
```

```
; *****
```

```
; Our data section. Here we declare our strings for our console message
```

```
; *****
```

```
.data
```

```
    X dw ?
```

```
    Y dw ? ;Результат
```

```
    A dw ? ;Y не меняется, поэтому и назвал A
```

```
    N dw ?
```

```
    tmp dw ?
```

```

input_buf db 06,00,5 dup(?)    ; Буфер для ввода числа

in_X db 0ah, 0dh, 'Input X :$' ; Буферы для переменных

in_A db 0ah, 0dh, 'Input Y :$'

in_N db 0ah, 0dh, 'Input N :$'

answer db 7 dup(?), '$'        ;ВВОД

messout db 0dh, 0ah, 'Result : $' ;ВЫВОД

; *****

; Our executable assembly code starts here in the .code section

; *****

.code

.startup

beg: jmp start

include bin2str.asm    ; Подключение программ для преобразования входных
include str2bin.asm    ; И выходных данных

;ВВОД

start:    lea dx,in_X    ; загрузка адреса буфера

mov ah,09h ; в AH - номер функции вывода строки

int 21h ; вызов прерывания DOS для вывода строки на экран

lea dx,input_buf ; в dx - адрес буфера ввода

mov ah,0ah ; в AH - номер функции ввода числа с клавиатуры

int 21h ; вызов прерывания ввода числа с клавиатуры

;преобразование строки в число

mov bx,dx ; перегрузить в bx адрес буфера

inc bx ; увеличить адрес на единицу

```

call str2bin ; обратиться к подпрограмме преобразования

mov X,ax ; запомнить число в переменной A

;ВВОД

lea dx,in_A ; загрузка адреса буфера

mov ah,09h ; в AH - номер функции вывода строки

int 21h ; вызов прерывания DOS для вывода строки на экран

lea dx,input_buf ; в dx - адрес буфера ввода

mov ah,0ah ; в AH - номер функции ввода числа с клавиатуры

int 21h ; вызов прерывания ввода числа с клавиатуры

;преобразование строки в число

mov bx,dx ; перегрузить в bx адрес буфера

inc bx ; увеличить адрес на единицу

call str2bin ; обратиться к подпрограмме преобразования

mov A,ax ; запомнить число в переменной A

;ВВОД

lea dx,in_N ; загрузка адреса буфера

mov ah,09h ; в AH - номер функции вывода строки

int 21h ; вызов прерывания DOS для вывода строки на экран

lea dx,input_buf ; в dx - адрес буфера ввода

mov ah,0ah ; в AH - номер функции ввода числа с клавиатуры

int 21h ; вызов прерывания ввода числа с клавиатуры

;преобразование строки в число

mov bx,dx ; перегрузить в bx адрес буфера

inc bx ; увеличить адрес на единицу

call str2bin ; обратиться к подпрограмме преобразования

mov N,ax ; запомнить число в переменной N

; $Y := P(Y, X^2) - P(Y, X-1) + P(Y, X)$, где $P(Z) = az^n - a^{n+1}$, При этом Y заносится как A , потому что не меняется параметр

```

mov Y,0      ;занести в Y 0

mov ax,X     ;в регистр заносится X

imul ax      ;x^2

push ax      ; поместить x^2 в стек

push A;поместить в стек A

push N;поместить в стек N

call pod     ;вызов подпрограммы для  $a*(x^2)^n - a^{n+1}$ 

add Y,ax     ;прибавить вычисленное выражение к Y

mov ax,X     ;в регистр заносится X

sub ax,1     ;X-1

push ax      ;поместить x-1 в стек

push A;поместить в стек A

push N;поместить в стек N

call pod     ;вызов подпрограммы для  $a*(x-1)^n - a^{n+1}$ 

sub Y,ax     ;вычесть вычисленное выражение из Y

mov ax,X     ;в регистр заносится X

push ax      ;поместить x в стек

push A;поместить в стек A

push N;поместить в стек N

call pod     ;вызов подпрограммы для  $a*x^n - a^{n+1}$ 

add Y,ax     ;прибавить вычисленное выражение к Y

mov ax,Y     ;в регистр заносится Y

mov Y,ax     ;результат поместить в Y

```

lea bx,answer ; поместить в BX адрес буфера для символьного представления

call bin2str ; обратиться к подпрограмме преобразования

lea dx,messout ; в dx - адрес буфера вывода

mov ah,09h ; в ah - номер функции вывода на экран

int 21h ; обратиться к функции вывода через 21 прерывание

lea dx,answer+1

mov ah,09h

int 21h

int 20h

;подпрограмма для расчета выражения $P(Z)=az^n-a^{n+1}$

pod proc near

parX equ [bp+8] ;через

parA equ [bp+6] ;директиву EQU присвоили

parN equ [bp+4] ;имена параметрам в стеке

push bp ;стандартная

mov bp,sp ;входная последовательность

;start

mov cx,parN ;в регистр заносится счетчик цикла

mov ax,1 ;в регистр заносится 1

;

cycl1: push cx ;поместить счетчик цикла в стек

mov bx,parX ;в регистр заносится параметр x

imul bx ;

pop cx ;восстановить из стека счетчик

loop cycl1;x^n in ax

mov bx,parA ;в регистр заносится параметр A

imul bx;a*x^n in ax

```

mov tmp,ax;a*x^n in tmp

mov cx,parN ;в регистр заносится счетчик цикла

mov ax,1 ;в регистр заносится 1

;a^n

cycl2: push cx ;поместить счетчик цикла в стек

mov bx,parA ;в регистр заносится параметр A

imul bx ;a^n

pop cx ;восстановить из стека счетчик

loop cycl2;a^n in ax

mov bx,tmp ;поместить в регистр a*x^n

sub bx,ax;a*x^n-a^n

add bx,1 ;a*x^n-a^n+1

mov ax,bx ;поместить a*x^n-a^n+1 в регистр

;end of podprog

pop bp ;ввосстановили bp

ret 6 ;возврат с очисткой стека от 3х параметров

pod endp

End

```

Тесты:

X	Y(A)	N	Y
2	2	2	35

```

Input X :2
Input Y :2
Input N :2
Result : 35
C:\>

```