Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра компьютерных систем в управлении и проектировании (КСУП)

ОТЧЕТ

Контрольная работа № 2

по дисциплине
«Информатика»

Выполнил студент:

специальности:

2023 г.

КОНТРОЛЬНАЯ РАБОТА № 2
РАЗРАБОТКА ПРОГРАММЫ НА АССЕМБЛЕРЕ

Вариант 11

Цель работы

Целью выполнения данной работы является комплексная проверка навыков программирования на языке ассемблера.

Результаты работы представляются в виде совокупности следующих документов:

1) титульный лист;
2) дерево подпрограмм;
3) файловая структура программы;
4) блок-схемы алгоритмов процедур;
5) исходный файл (файлы) программы;
6) загрузочный модуль программы.

Основным требованием к блок-схемам алгоритмов процедур является выполнение требований структурного программирования.

Основным требованием к исходным модулям (файлам) программы является наличие комментариев.

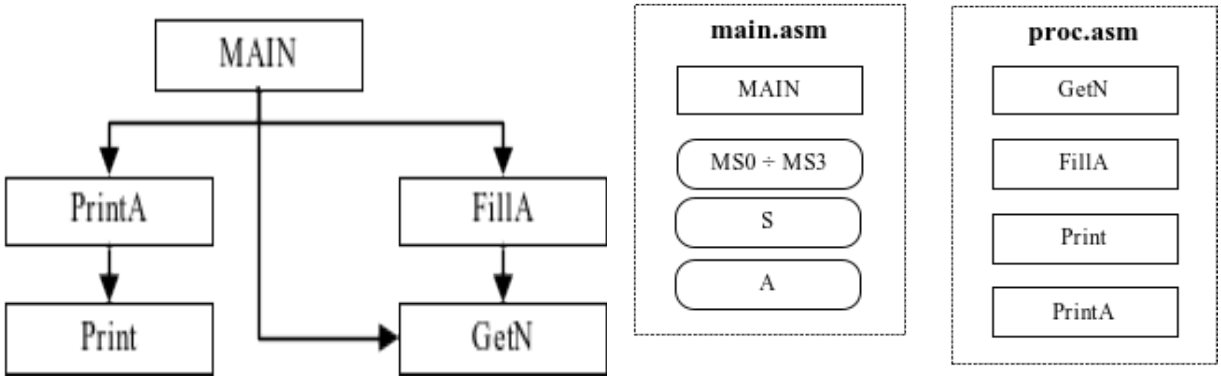
Дерево подпрограмм, файловая структура программы и блок-схемы процедур представляются в виде файлов, полученных с помощью текстового редактора Word. Остальные документы представляются в виде файлов с расширениями .asm и .com и помещаются в папку CONTR2.

Задание

По запросу программы пользователь вводит с клавиатуры целое положительное десятичное число N. По следующему запросу он вводит с клавиатуры N целых трехзначных положительных десятичных чисел, разделенных пробелами.

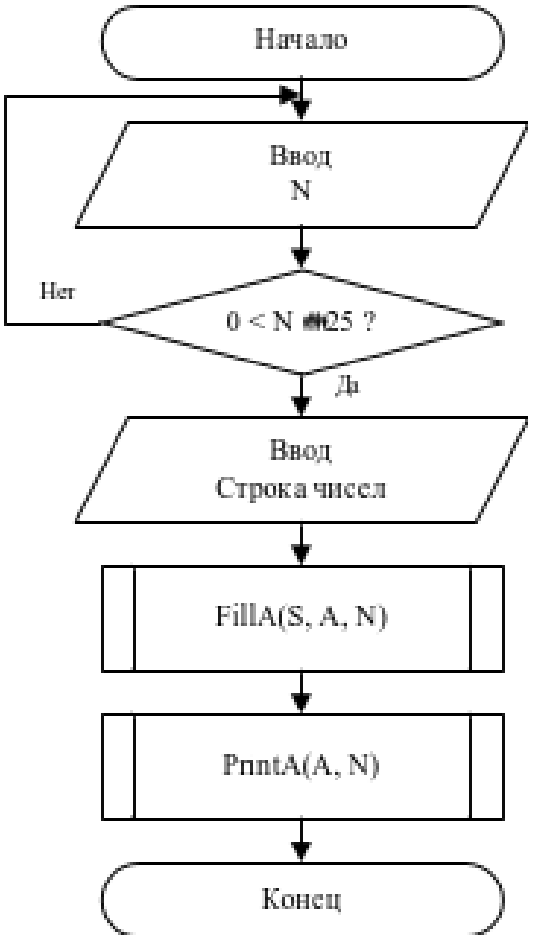
Программа выводит последовательность этих же чисел, но записанных в обратном порядке и в восьмеричной системе счисления.

Дерево подпрограмм и файловая структура программы

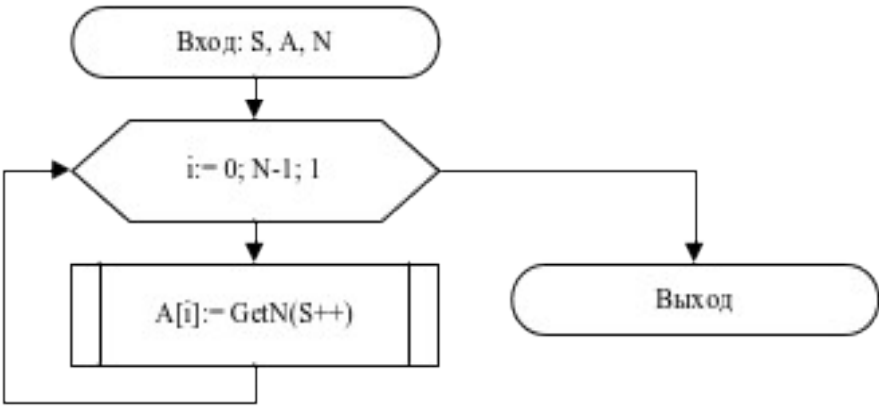


Блок-схемы алгоритмов процедур

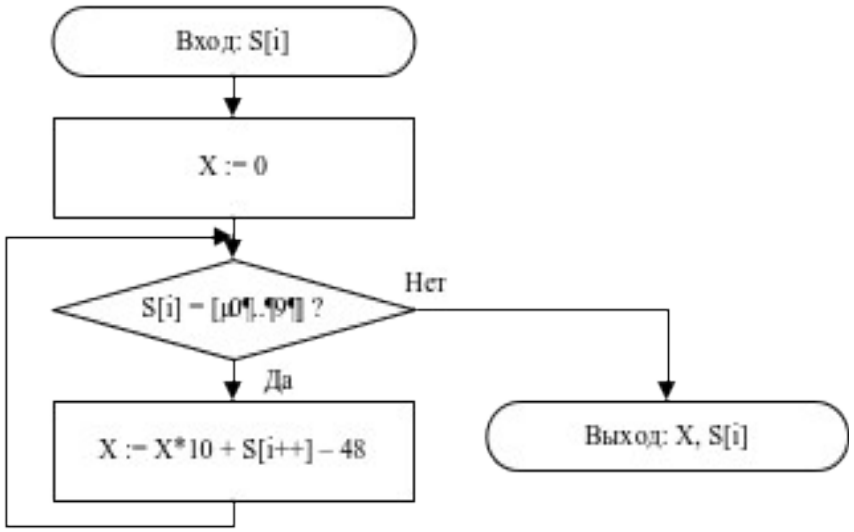
Главная процедура MAIN:



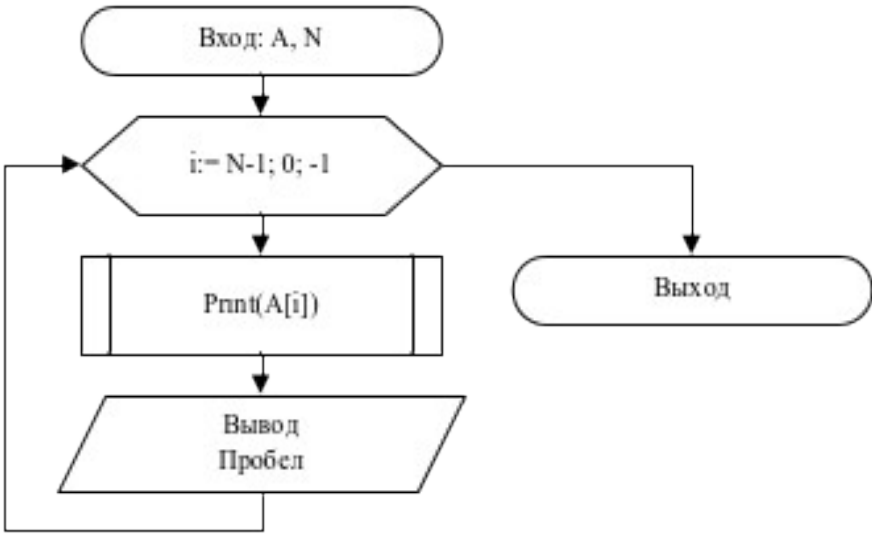
Процедура заполнения массива числами из строки FillA:



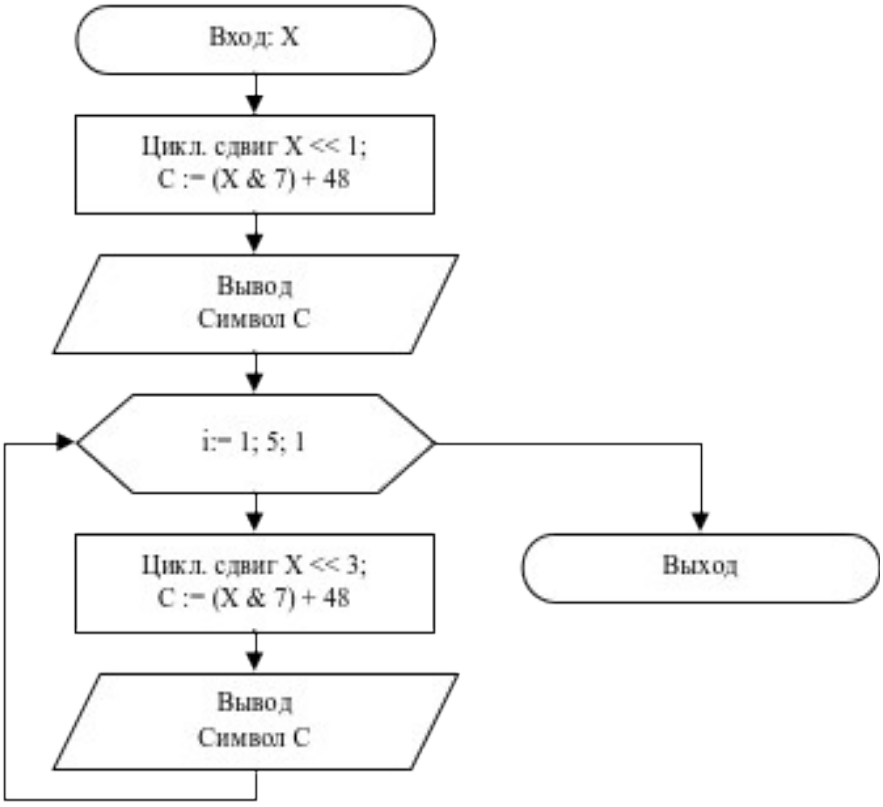
Процедура перевода части строки в число GetN:



Процедура вывода массива PrintA:



Процедура вывода числа в 8-ой системе счисления Print:



Листинг файла MAIN.ASM

```
      [ORG 100h]
MAIN:                 ; главная процедура
      ; вывод сообщения на ввод N
      MOV DX, MS0
      MOV AH, 09
      INT 21h
      ; ввод числа в буфер
      MOV CX, 4
      MOV DX, S
      MOV BX, 1
      MOV AH, 3Fh
      INT 21h
      ; получение численного значения
      MOV SI, S
      CALL GetN
      ; проверка введенного числа N
      AND AX, AX
      JZ MAIN         ; если введено количество чисел 0 - повтор ввода
      CMP AX, 25
      JA MAIN         ; если кол-во чисел N > 25 - повтор ввода
      PUSH AX         ; поместить количество чисел в стек
      ; вывод сообщения для ввода строки чисел
      MOV DX, MS1
```

```
        MOV AH, 09
        INT 21h
        ; ввод строки чисел в буфер
        MOV DX, S
        MOV CX, 148
        MOV BX, 1
        MOV AH, 3Fh
        INT 21h
        POP CX          ; CX <- кол-во чисел
        ; выборка из буфера чисел в массив
        CLD             ;  автоинкремент  при  работе  с  цепочечными
командами
        MOV SI, S       ; загрузить в SI адрес начала строки в буфере
        MOV DI, A       ; загрузить в DI адрес первого элемента массива
        CALL FillA      ; вызов подпрограммы заполнения массива
        ; вывод массива
        MOV DX, MS2     ; вывод сообщения
        MOV AH, 09
        INT 21h
        MOV SI, A       ; загрузить в SI адрес массива
        CALL PrintA     ; вызов подпрограммы вывода массива
        ; вывод сообщения - Конец работы
        MOV DX, MS3
        MOV AH, 09
        INT 21h
        ; ждать ввода
        XOR AX, AX
        INT 16h
        ; выход в DOS
        INT 20h


        ; подключение модуля процедур
        %INCLUDE "PROC.ASM"

; сегмент данных
SECTION DATA
        ; выводимые сообщения
        MS0 DB 13, 10, 'Введите число N = [1..25]: $'
        MS1 DB 13, 10, 10, 'Введите последовательность 3-значных положит.
чисел разделенных пробелом:', 13, 10, '$'
        MS2 DB 13, 10, 10, 'Эти же числа, записанные в обратном порядке и
в 8-ной с.с.:', 13, 10, '$'
        MS3 DB 13, 10, 10, 'Нажмите клавишу для выхода...$'
        S RESB 150      ; буфер под строку
        A RESW 25       ; массив
```

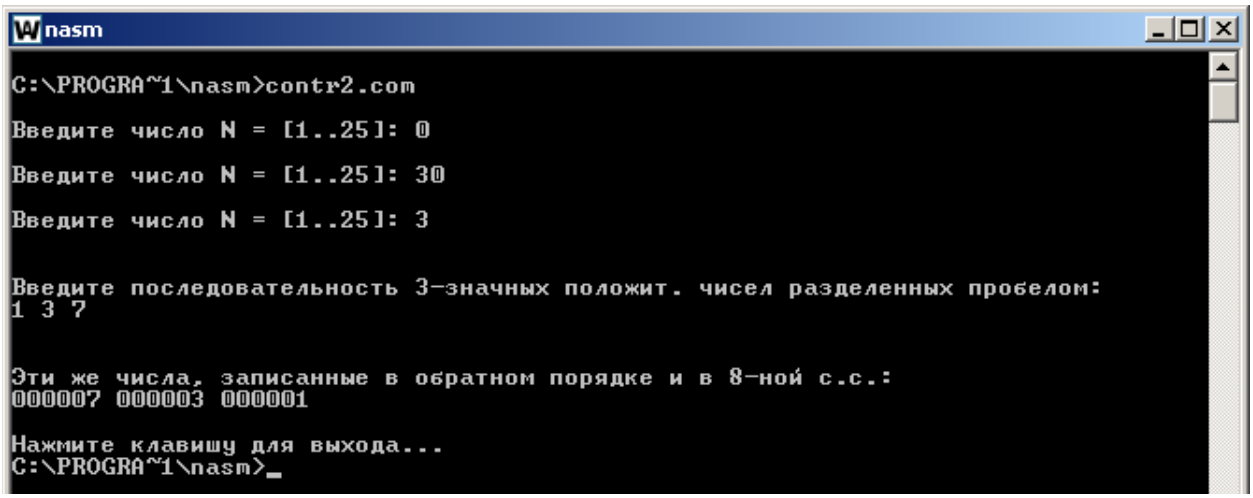
Листинг файла PROC.ASM

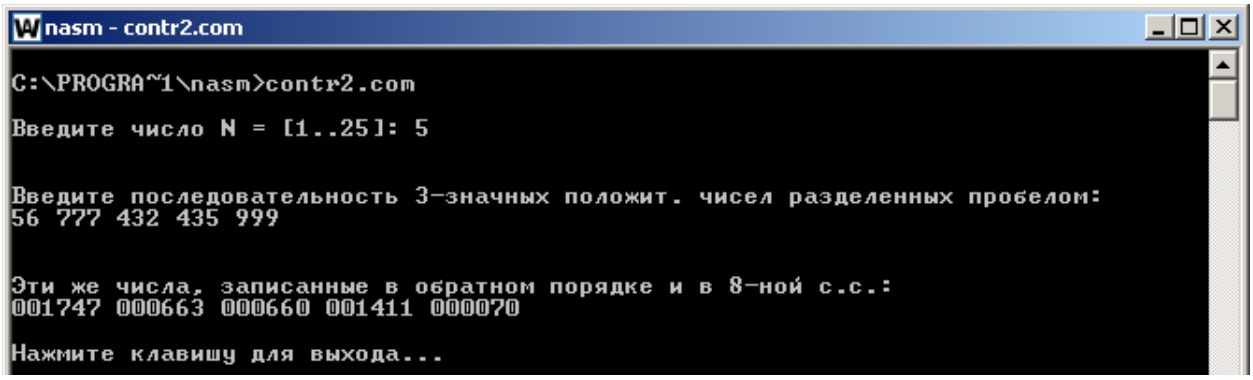
```
; Процедура заполнения массива числами
; Вход: AX - число
;     SI - адрес начала буфера
;     DI - адрес массива
;     CX - кол-во чисел
FillA:
      PUSH CX          ; помещаем в стек значения используемых регистров

_fa:
      CALL GetN        ; преобразуем строку в число
      STOSW            ; сохраняем его в массив и переходим к следующему
элементу
      INC SI           ; пропускаем разделитель
      LOOP _fa         ; повторяем пока преобр-ны не все числа из строки

      POP CX           ; восстанавливаем CX из стека
      RET

; Процедура вывода числа в 8-ой системе сч.
; Вход: AX - число
Print:
      PUSH AX          ; помещаем в стек значения используемых регистров
      PUSH BX
      PUSH CX
      PUSH DX

      MOV BX, AX       ; BX <- число
      MOV AH, 02       ; AH <- функция вывода символа на экран
      MOV CX, 0503H    ; CH <- кол-во сдвигов, CL - сдвиг на 3 разряда
      ROL BX, 1        ; сдвиг BX на 1 разряд
      MOV DL, '0'      ; DL <- символ, соответствующий старшей
      ADC DL, 0        ; цифре
      INT 21h          ; вывод цифры на экран
_p:
      ROL BX, CL       ; сдвиг BX на 3 разряда
      MOV DL, BL       ; AX <- Выход от деления
      AND DL, 7        ; DL - цифра
      ADD DL, 48       ; DL <- ASCII код цифры
      INT 21h          ; вывод цифры на экран
      DEC CH           ; уменьшаем счетчик сдвигов
      JNZ _p           ; повторяем пока CH > 0

      POP DX           ; извлекаем регистры из стека
      POP CX
      POP BX
```

```
      POP AX
      RET             ; возврат из подпрограммы


;Процедура вывода массива чисел в обратном порядке
; Вход: SI - адрес массива
;     CX - кол-во чисел
PrintA:
      PUSH CX         ; помещаем в стек значения используемых регистров
      PUSH SI
      ADD SI, CX      ; SI = SI + CX
      ADD SI, CX      ; адрес за последним числом
_pa:
      SUB SI, 2       ; SI <- SI - 2
      MOV AX, [SI]    ; AX <- текущий эл. массива
      CALL Print      ; вызов п/п вывода числа на экран
      MOV DL, " "     ; загрузка в DL кода пробела
      MOV AH, 02
      INT 21h         ; вывод пробела
      LOOP _pa        ; цикл вывода массива

      POP SI          ; восстанавливаем значения регистров из стека
      POP CX
      RET

; Процедура перевода части строки в число
; Вход:  в SI начальный адрес в буфере
; Выход: AX - число
GetN:
      PUSH DX         ; помещаем в стек значения используемых регистров
      PUSH BX

      XOR AX, AX ; AX <- 0
      MOV BX, AX ; BX <- 0

_gn0:
      MOV BL, [SI]    ; BL <- код символа из буфера
      SUB BL, 48 ;
      JC _gn1         ; если код меньше "0" конец перевода
      CMP BL, 9       ; проверка на цифру
      JA _gn1         ; если не цифра - конец перевода

      MOV DX, 10
      MUL DX          ; умножаем AX на 10
      ADD AX, BX      ; увеличиваем число на текущую цифру
      INC SI          ; переход к очередному символу буфера
      JMP _gn0        ; повтор тела цикла

_gn1:
      POP BX          ; восстанавливаем значения регистров из стека
```

```
POP DX
RET             ; возврат из подпрограммы
```

Результаты выполнения





В ходе выполнения контрольной работы была разработана программа согласно варианту задания. Скриншоты окна выполнения программы доказывают правильность и полноту решения задачи и достижения выносимых на контрольную работу целей.