Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет

информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина: Конструирование программ

Отчёт по лабораторной работе №7

ЦЕЛОЧИСЛЕННЫЕ АРИФМЕТИЧЕСКИЕ ОПЕРАЦИИ.

ОБРАБОТКА МАССИВОВ ЧИСЛОВЫХ ДАННЫХ

Студент Кончик Д.С.

Принимающий Романюк М.В.

Минск 2022

**СОДЕРЖАНИЕ**

[1 Цель работы 3](#_Toc118909172)

[2 Задание 4](#_Toc118909173)

[3 Результат выполнения 8](#_Toc118909174)

[4 Вывод 11](#_Toc118909175)

**1 ЦЕЛЬ РАБОТЫ**

1. Ознакомиться с арифметичскими операциями над целочисленными данными, обработкой массивов чисел;
2. ознакомиться с правилами оформления ассемблерных процедур.

2 ЗАДАНИЕ

Вариант 12. Ввести массив целых чисел размерностью 30 элементов. Построить гистограмму для заданного диапазона чисел.

Листинг 1 ­– Код программы

; Распечатать строку [Макрос]

printStr macro str

push ax

push dx

mov ah,09h ; Номер функции DOS

lea dx,str ; Смещение str в DX

int 21h ; 21 прерывание

pop dx

pop ax

endm

; Модель памяти (код в 1 сегменте, данные и стек в DGROUP)

.model small

; Сегмент кода

.code

; Начало прогарммы

START:

mov ax,@data ; @data - идентификатор сегмента данных

mov ds,ax ; Адрес сегмента данных — в DS

mov es,ax ; Адрес сегмента данных — в ES

; Задание максимальной длины для вводимого числа

lea di,strNumber ; Смещение строки в DI

mov al,06d ; Максимальное число символов для ввода

stosb ; AL в ES:DI

; Вывод "какой формат числел"

printStr strFormat

printStr toNewLine

printStr toNewLine

; Заполнение массива

mov cx,array\_size ; CX = array\_size

lea si,array ; SI на начало массива

inputWhile:

push cx ; Сохранить CX

push si ; Сохранить SI

call enterNumber

printStr toNewLine

cmp isValidNumber,01d

je continue

jmp end

; Ввод валидный

continue:

pop si ; Восстановить SI

pop cx ; Восстановить CX

mov dx,number

mov [si],dx ; Записать число в массив

add si,2d

; Условие выхода из цикла

dec cx

cmp cx,0d

jnz inputWhile

; Рисование гистограммы

mov cx,0h ; CX = 0

lea si,array ; SI на начало массива

printStr toNewLine

for:

mov ax,[si] ; AX = текущий элемент массива

mov dx,0h

mul max\_symb

div max\_item ; AX = число символов для столбца

push cx

; Вывод символов

printStr symbStart ; Символы начала столбика

mov cx,ax

jcxz noSymbols

symbolsOutput:

printStr symb

; Условие завершения цикла

dec cx

cmp cx,0h

jnz symbolsOutput

noSymbols:

pop cx

printStr toNewLine

add si,2d

; Условие завершения цикла

inc cx

cmp cx,array\_size

jnz for

; Конец программы

end:

mov ax,4C00h ; 4C (выход из программы) в AH

; 00 (успешное звершение) в AL

int 21h ; Функция DOS "Завершить программу"

; Процедура – ввод числа

; [Результат занесется в dx]

enterNumber proc near

; Ввод числа в строку

mov ah,0Ah ; Функция DOS (считать строку символов из STDIN в буфер)

lea dx,strNumber ; Смещение строки в DX

int 21h

; Вычисление длины введенного числа

lea si,strNumber ; Смещение strNumber в SI

add si,01d ; SI++

mov ax,0h ; Занулить AX

lodsb ; DS:SI в AL

mov cx,ax ; CX = длина введенного числа

; Длина введенного слова 0

cmp cx,0h

jz strInvalid

mov number,0h ; Число будет формироваться в number

mov ax,01d ; AX – степень десятки

while:

mov bx,0h

mov si,cx

add si,01d ; SI – смещение на текущий символ числа

mov bl,strNumber[si] ; BX – ASCII код текущего символа

; Проверка на то, является ли символ цифрой 0..9

cmp bl,48d

jl strInvalid; < 48

cmp bl,57d

jg strInvalid; > 57

sub bl,48d ; BX = текущая цифра

push ax ; Сохранение AX

clc ; Очистить флаг переноса

mul bx ; AX = текущая цифра \* 10 в степени

jc overflow ; Если есть флаг переноса от mul

add number,ax ; Добавляем это в формируемое число

jc overflow ; Если есть флаг переноса от add

pop ax ; Восстановление AX

mul ten ; AX \*= 10

; Условие выхода из цикла

dec cx

cmp cx,0

jne while

; Поиск максимального элемента

mov dx,max\_item

cmp number,dx

ja new\_max ; if number > max\_item

ret

new\_max:

mov dx,number

mov max\_item,dx

ret

; Переполнение

overflow:

pop ax ; Чтобы вернуть SP на IP

mov isValidNumber,0d

printStr toNewLine

printStr strOverflow

ret

; Невалидная строка

strInvalid:

mov isValidNumber,0d

printStr toNewLine

printStr strInvalidFormat

ret

enterNumber endp

; Сегмент данных

.data

strNumber db 10 DUP(?) ; Строка, в которую вводим число

number dw 0d ; Введенное число (высчитывается)

strOverflow db 'Entered number is more than 65535$'

strInvalidFormat db 'Entered number is invalid$'

strFormat db 'Enter non-negative hexadecimal numbers in',0Dh,0Ah,'the decimal notation separated by an enter$'

isValidNumber db 01d

ten dw 10d

max\_symb dw 50d ; Максимальное количество символов для гистограммы

symb db '#$' ; Символ для заполнения столбика

symbStart db '> $' ; Символ для начала столбика

array\_size dw 30 ; Заполняемое число элементов массива

array dw 40 DUP(?) ; Массив чисел

max\_item dw 0 ; Максимальный элемент массива

toNewLine db 0Ah, 0Dh, '$' ; Символы перехода на новую строку (v,<)

; Сегмент стека (256 байт)

.stack 100h

; Конец программы

end START

3 РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ

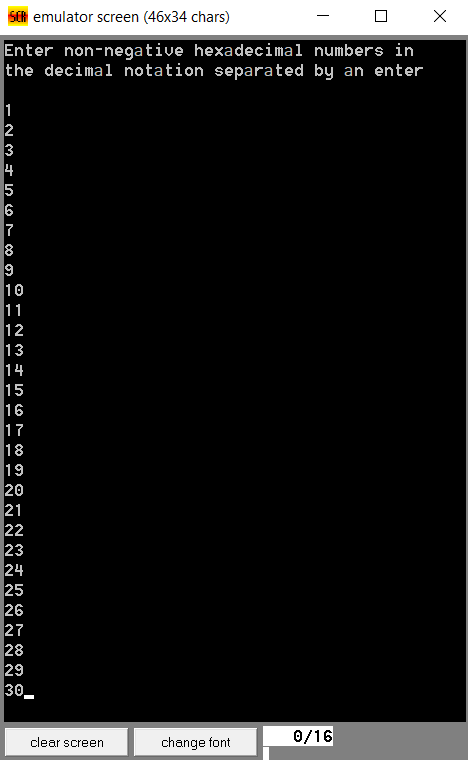


Рисунок 1 – Ввод массива чисел

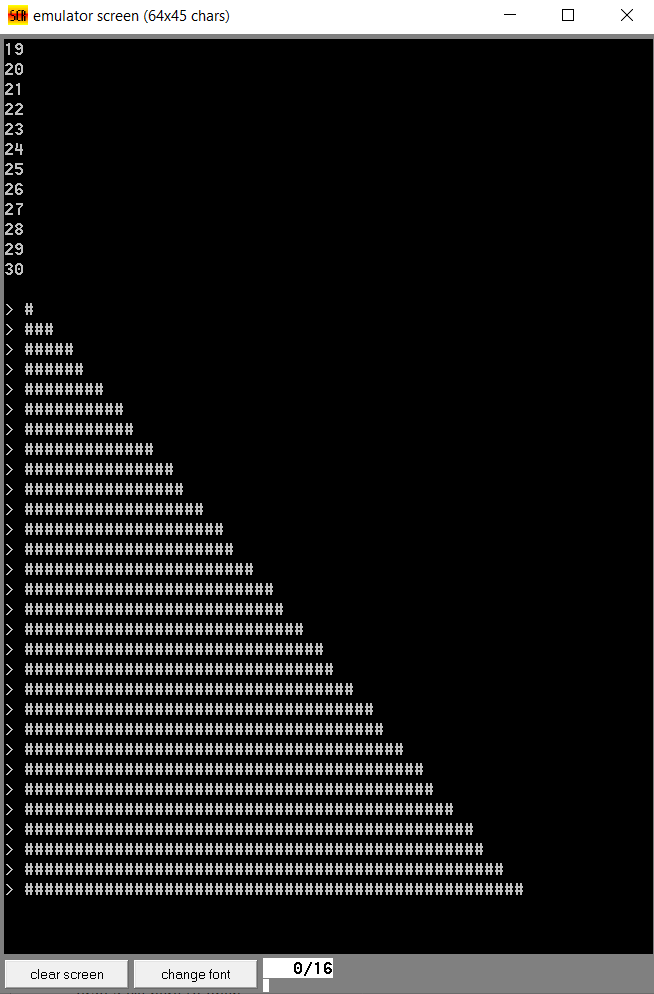


Рисунок 2 – Результат работы программы

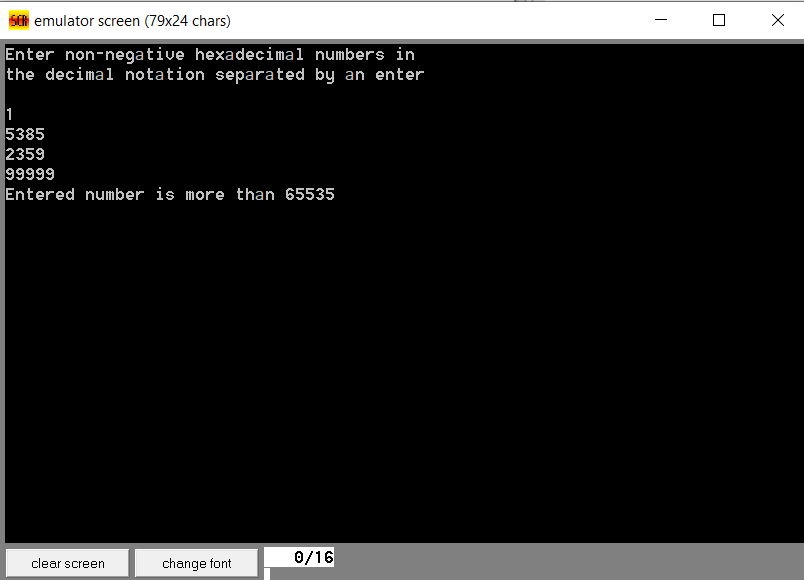


Рисунок 3 – Обработка некорректного ввода (переполнение)

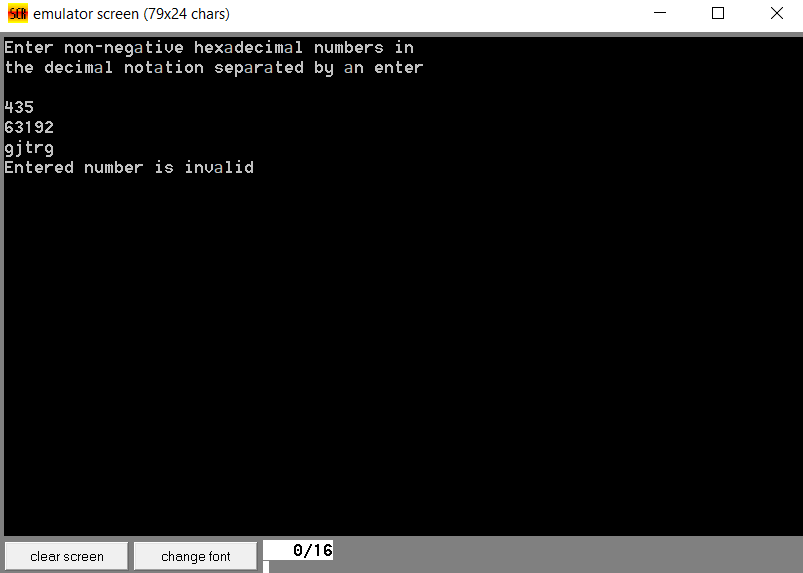


Рисунок 4 – Обработка некорректного ввода (введено не число)

4 ВЫВОД

Таким образом, была созданна программа, принимающая на ввод 30 элементов массива и строящая гистограмму массива (большему числу соответствует большее число символов «#»). Также программа обрабатывает некорректный ввод: ввод числа, большего, чем может поместиться в 16 разрядов; последовательности символов, не являющимися числом.