|  |  |  |
| --- | --- | --- |
| Министерство образования Республики Беларусь  Учреждение образования  Белорусский Государственный Университет Информатики и Радиоэлектроники | | |
| Факультет компьютерного проектирования | | |
| Кафедра инженерной психологии и эргономики | | |
| Работа с массивами данных | | |
| Выполнил  студент гр. 810101  Лисенков Д. Ю. |  | Проверила  Киринович И. Ф. |
| Минск 2021 | | |

# Цель работы

Приобретение навыков работы с массивами данных. Изучение способов адресации данных в массиве.

# ХОД ВЫПОЛНЕНИЯ РАБОТЫ

Задание:

Создать линейный массив из n целых чисел. Число n запрашивается с клавиатуры. Элементы массива, состоящие из положительных, отрицательных чисел и нулей вперемешку вводятся с клавиатуры. Создать новый массив, в котором числа будут упорядочены след. образом: положительные, нули и отрицательные, в порядке их следования в исходном массиве.

Решить задачу двумя способами: используя прямую адресацию с индексированием, косвенная индексация.

Результаты выполнения программы вывести на экран.

Листинг кода:

org 100h

mov dx, offset str1 ;поместить в регистр dx строку str1

mov ah,09 ;поместить в регистр ah значение 09

int 21h ;прерывание 21h

Call Input ;вызов процедуры Input

mov n,bx ;поместить в переменную n значение

;регистра bx

mov si,0 ;поместить в регистр si значение 0

mov cx,0 ;поместить в регистр cx значение 0

mov dx, offset str2 ;поместить в регистр dx строку str2

mov ah,09 ;поместить в регистр ah значение 09

int 21h ;прерывание 21h

begin: ;метка begin

cmp cx,n ;сравнить значение регистра cx с n

je exit ;переход если равно на метку exit

Call Input ;вызов процедуры Input

mov mass[si],bx ;поместить в массив значение регистра bx

add si,2 ;прибавить к значению регистра si

;значение 2

inc cx ;увеличить значение регистра cx на 1

jmp begin ;переход на метку begin

exit: ;метка exit

mov dx, offset str3 ;поместить в регистр dx строку str3

mov ah,09 ;поместить в регистр ah значение 09

int 21h ;прерывание 21h

mov si,0 ;поместить в регистр si значение 0

mov di,0 ;поместить в регистр di значение 0

Call IFB0 ;вызов процедуры IFB0

mov si,0 ;поместить в регистр si значение 0

mov di,0 ;поместить в регистр di значение 0

Call IFR0 ;вызов процедуры IFR0

mov si,0 ;поместить в регистр si значение 0

mov di,0 ;поместить в регистр di значение 0

Call IFM0 ;вызов процедуры IFM0

Call OutMass ;вызов процедуры OutMass

mov dx, offset str4 ;поместить в регистр dx строку str4

mov ah,09 ;поместить в регистр ah значение 09

int 21h

lea si,mass ;поместить адрес mass в si

lea di,newMass ;поместить адрес newMass в di

Call IFB0 ;вызов процедуры IFB0

lea si,mass ;поместить адрес mass в si

lea di,newMass ;поместить адрес newMass в di

Call IFR0 ;вызов процедуры IFR0

lea si,mass ;поместить адрес mass в si

lea di,newMass ;поместить адрес newMass в di

Call IFM0 ;вызов процедуры IFM0

Call OutMass ;вызов процедуры OutMass

ret ;вернуть

n dw ? ;объявление переменной n

mass dw n dup(?) ;объявление массива mass

newMass dw n dup(?) ;объявление массива newMass

str1 db "Vvedite n = $" ;объявление строки str1

str2 db 10,13,"Vvedite elementi massiva: $" ;объявление строки str2

str3 db 10,13,"Priamaia adresaciia s indeksirovaniem $" ;объявление

;строки str3

str4 db 10,13,"Kosvennaia adresaciia $" ;объявление строки str4

str5 db 10,13,"Novii massiv: $" ;объявление строки str5

IFB0 PROC ;процедура IFB0

mov cx,0 ;поместить в регистр cx значение 0

b2: ;метка b2

cmp cx,n ;сравнить значение регистра cx и значение

;переменной n

je ex2 ;переход если равно на метку ex2

cmp mass[si],0 ;сравнить значение в массиве с 0

jbe lb2 ;переход если меньше или равно

;на метку lb2

mov ax,mass[si] ;поместить в регистр ax значение элемента

;массива

mov newMass[di],ax ;поместить в элемент нового массива

;значение регистра ax

add si, 2 ;прибавить к значению регистра si

;значение 2

add di, 2 ;прибавить к значению регистра di

; значение 2

inc cx ;увеличить значение регистра cx на 1

lb2: ;метка lb2

jmp b2 ;переход на метку b2

ex2: ;метка ex2

ret ;вернуть

IFB0 ENDP ;конец процедуры IFB0

IFR0 PROC ;процедура IFR0

mov cx,0 ;поместить в регистр cx значение 0

b3: ;метка b3

cmp cx,n ;сравнить значение регистра cx и значение

;переменной n

je ex3 ;переход если равно на метку ex3

cmp mass[si],0 ;сравнить значение в массиве с 0

jae lb3 ;переход если меньше или больше

;на метку lb2

mov ax,mass[si] ;поместить в регистр ax значение элемента

;массива

mov ds:newMass[di],ax ;поместить в элемент нового массива

;значение регистра ax

add si, 2 ;прибавить к значению регистра si

;значение 2

add di, 2 ;прибавить к значению регистра di

;значение 2

inc cx ;увеличить значение регистра cx на 1

lb3: ;метка lb3

jmp b3 ;переход на метку b3

ex3: ;метка ex3

ret ;вернуть

IFR0 ENDP ;конец процедуры IFR0

IFM0 PROC ;процедура IFM0

mov cx,0 ;поместить в регистр cx значение 0

b4: ;метка b4

cmp cx,n ;сравнить значение регистра cx и значение

;переменной n

je ex4 ;переход если равно на метку ex4

cmp mass[si],0 ;сравнить значение в массиве с 0

jne lb4 ;переход если больше или равно

;на метку lb2

mov ax,mass[si] ;поместить в регистр ax значение элемента

;массива

mov newMass[di],ax ;поместить в элемент нового массива

;значение регистра ax

add si, 2 ;прибавить к значению регистра si

;значение 2

add di, 2 ;прибавить к значению регистра di

;значение 2

inc cx ;увеличить значение регистра cx на 1

lb4: ;метка lb4

jmp b4 ;переход на метку b4

ex4: ;метка ex4

ret ;вернуть

IFM0 ENDP ;конец процедуры IFM0

Input PROC ;процедура Input

push ax ;поместить в стек значение регистра ax

push cx ;поместить в стек значение регистра cx

push dx ;поместить в стек значение регистра dx

mov bx,0000 ;поместить в регистр bx значение 0000

mov cx,0000 ;поместить в регистр cx значение 0000

mov dx,1 ;поместить в регистр dx значение 1

mov ah,01 ;поместить в регистр ah значение 01

l1: ;метка l1

mov ah,01 ;поместить в регистр ah значение 01

int 21h ;прерывание 21h

cmp al,0DH ;сравнить значение в регистре al с 0DH

je l2 ;переход если равно на метку l2

cmp al,'-' ;сравнить значение в регистре al с '-'

je l3 ;переход если равно на метку l3

sub al,30h ;вычесть от значения регистра al

;значение 30h

l3: ;метка l3

mov ah,0 ;поместить в регистр ah значение 0

push ax ;поместить в стек значение регистра ax

inc cx ;увеличить значение регистра cx на 1

jmp l1 ;переход на метку l1

l2: ;метка l2

pop ax ;достать из стека значение и поместить

;его в ax

cmp al,'-' ;сравнить значение в регистре al с '-'

jne l4 ;переход если не равно на метку l4

neg bx ;изменить знак регистра bx

jmp l5 ;переход на метку l5

l4: ;метка l4

mul dl ;умножить значение регистра al

;на значение регистра dl

add bx,ax ;прибавить к регистру bx

;значение регистра ax

mov ax,dx ;поместить в регистр ax значение

;регистра dx

mov dx,000AH ;поместить в регистр dx значение 000AH

mul dl ;умножить значение регистра al

;на значение регистра dl

mov dx,ax ;поместить в регистр dx значение

;регистра ax

loop l2 ;цикл loop

l5: ;метка l5

pop dx ;достать из стека значение и поместить

;его в dx

pop cx ;достать из стека значение и поместить

;его в cx

pop ax ;достать из стека значение и поместить

;его в ax

ret ;вернуть

Input ENDP ;конец процедуры Input

OutNum proc ;процедура OutNum

aam ;исправляет BCD-значение после

;умножения

add ax,3030h ;прибавить к регистру ax значения 3030h

mov dl,ah ;поместить в регистр dl значение

;регистра al

mov dh,al ;поместить в регистр dh значение

;регистра al

mov ah,02 ;поместить в регистр ah значение 02

int 21h ;прерывание 21h

mov dl,dh ;поместить в регистр dl значение

;регистра dh

int 21h ;прерывание 21h

ret ;вернуть

OutNum endp ;конец процедуры OutNum

OutMass proc ;процедура OutMass

mov dx, offset str5 ;поместить в регистр dx строку str5

mov ah,09 ;поместить в регистр ah значение 09

int 21h ;прерывание 21h

begin2: ;метка begin2

cmp cx,n ;сравнить значение в регистре cx c n

je exit2 ;переход если равно на метку exit2

mov ax,mass[si] ;поместить в регистр ax значение

;элемента массива

Call OutNum ;вызов процедуры OutNum

add si,2

inc cx ;увеличить значение регистра cx на 1

jmp begin 2 ;переход на метку begin2

exit2: ;метка exit2

ret ;вернуть

OutMass endp ;конец процедуры OutMass

Результаты выполнения программы:

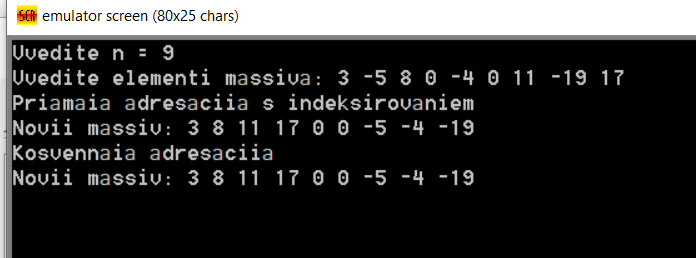


Рис. 1 – Результат выполнения программы

# Вывод

Приобрел навыки работы с массивами данных. Изучил способы адресации данных в массиве.