ГУАП

КАФЕДРА № 34

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Старший преподаватель |  |  |  | Жиданов К.А. |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ |
|  |
| по курсу: |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. | 3145 |  |  |  | Пьянов Ю.Д. |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2021

**Задание 1:** написать программу, реализующую заданный алгоритм:

Посчитать количество четных элементов в массиве.

**Ход выполнения:**

1. Чтобы определить количество четных элементов в массиве, достаточно обработать каждый элемент массива и сравнить остаток от деления этого элемента на 2 с нулём, а затем
2. Регистры для ввода и вывода данных:

EBX – Хранение адреса массива

DL – Хранение количества четных элементом массива

1. Заносим программу в отладчик
2. Выполняем программу, подавая на вход различные тестовые наборы, после каждого запуска убеждаясь, что результат работы программы является корректным и совпадает с полученным при выполнении вручную.

**Код программы:**

LEA ESI, [a]

MOV ECX, 8

MOV BL, 2

next:

MOV AL,[ESI]

DIV BL

CMP AH, 0

JZ summ

next1:

INC ESI

LOOP next

JMP quit

summ:

XOR AH,AH

INC DL

JMP next1

quit:

ret

**Задание 2:** написать программу, реализующую доступ к упакованному массиву с заданной разрядностью элементов: 6 бит

**Ход выполнения:**

1. Выбираем массив, элементы которого не превышают 63, например [32, 47, 5, 18, 61, 50, 24, 12]
2. Переводим числа в двоичную систему, а затем записываем биты от младшего к старшему.
3. Объединяем числа по 8 элементов и переводим в 16-ричную СС
4. Регистры для входных и выходных значений:

ESI – Адрес массива

AL, Bl – Элементы массива

1. Заносим программу в отладчик
2. Выполняем программу, проверяя каждый элемент поочерёдно, после каждого запуская убеждаясь, что программа работает корректно и результаты совпадают с полученными вручную.

**Код программы:**

LEA ESI,[a]

;извлекаем значение с индексом 0

MOV AL,[ESI]

MOV DL, 0x3F

AND AL,DL

;PRINT\_UDEC 1,AL

;извлекаем значение с индексом 1

MOV AL,[ESI + 1]

MOV DL, 0x0F

AND AL,DL

SHL AL,2

MOV BL,AL

MOV AL,[ESI]

MOV DL, 0xC0

AND AL,DL

SHR AL,6

OR BL,AL

PRINT\_UDEC 1,BL

;извлекаем значение с индексом 2

MOV AL,[ESI + 2]

MOV DL, 0x03

AND AL,DL

SHL AL,4

MOV BL,AL

MOV AL,[ESI + 1]

MOV DL, 0xF0

AND AL,DL

SHR AL,4

OR BL,AL

;PRINT\_UDEC 1,BL

;извлекаем значение с индексом 3

MOV AL,[ESI + 2]

MOV DL, 0xFC

AND AL,DL

SHR AL,2

;PRINT\_UDEC 1,AL

;извлекаем значение с индексом 4

MOV AL,[ESI + 3]

MOV DL, 0x3F

AND AL,DL

;PRINT\_UDEC 1,AL

;извлекаем значение с индексом 5

MOV AL,[ESI + 4]

MOV DL, 0x0F

AND AL,DL

SHL AL,2

MOV BL,AL

MOV AL,[ESI + 3]

MOV DL, 0xC0

AND AL,DL

SHR AL,6

OR BL,AL

;PRINT\_UDEC 1,BL

;извлекаем значение с индексом 6

MOV AL,[ESI + 5]

MOV DL, 0x03

AND AL,DL

SHL AL,4

MOV BL,AL

MOV AL,[ESI + 4]

MOV DL, 0xF0

AND AL,DL

SHR AL,4

OR BL,AL

;PRINT\_UDEC 1,BL

;извлекаем значение с индексом 7

MOV AL,[ESI + 5]

MOV DL, 0xFC

AND AL,DL

SHR AL,2

;PRINT\_UDEC 1,AL

ret

section .data

;packed array [32, 47, 5, 18, 61, 50, 24, 12]

a: DB 0xE0, 0x5B, 0x48, 0xBD, 0x8C, 0x31

**Задание 3:**

Написать программу, реализующую алгоритм сортировки: сортировка выбором.

**Ход работы:**

1. Сортировка выбором подразумевает нахождение минимального значения в текущем массиве и обмен этого значения со значением первой неотсортированной позиции.
2. В качестве тестового массива возьмём [5, 3, 6, 2, 1, 7, 4, 8], программ должна выдать [1,2,3,4,5,6,7,8]
3. Регистры для входных и выходных значений:

EBX – Адрес массива

ECX – Количество элементов массива

EDI – Адрес массива

AL, DL – Элементы массива

1. Заносим программу в отладчик
2. Выполняем программу и убеждаемся, что результат совпадает с вычисленным вручную.

**Код программы:**

LEA EBX, [a]

MOV ECX, 8

next:

PUSH EBX

PUSH ECX

MOV DL, [EBX]

MOV EDI, EBX

next1:

MOV AL, [EBX]

CMP AL, DL

JNL skip

MOV DL, AL

MOV EDI, EBX

skip:

INC EBX

LOOP next1

POP ECX

POP EBX

MOV AL, [EBX]

MOV [EDI], AL

MOV [EBX], DL

INC EBX

LOOP next

ret

section .data

a: DB 5, 3, 6, 2, 1, 7, 4, 8