

Лабораторная работа №7.
Использование массивов.
(Колпаков, Краснов, Шокоров)

Задание 1. Создайте, используя язык высокого уровня, массив $a[]$ из целых чисел.

Изучите расположение элементов массива (используя возможности языка высокого уровня или IDE).

1. Каков размер элемента (в байтах)?
2. Насколько отличаются адреса соседних элементов массива (в байтах)?
3. Одинаково или различно это расстояние для разных элементов массива?
4. Каков общий размер массива (в байтах)?
5. Как получить адрес элемента массива, зная его номер и адрес начала массива?

```
#include <iostream>

using std::cout;
using std::cin;
using std::endl;
int main(void) {
    int arr[10] = {1321, -412, 423, 693, 9124,
                  9542, 98542, 34123, 8432, 17432};
    int *firstElement = arr;
    int *secondElement = arr + 1;
    cout << endl << "An element size: " << sizeof(arr[0]) << endl;
    cout << "First element adress: " << firstElement << endl;
    cout << "Second element adress: " << secondElement << endl;
    cout << "Gap between elements: " << secondElement - firstElement <<
endl;
    cout << "The array size: " << sizeof(arr) << endl;
    return 0;
}
```

```
kdator@kdator:~/study$ ./main
An element size: 4
First element adres: 0x7ffecd0b7620
Second element adres: 0x7ffecd0b7624
Gap between elements: 1
The array size: 40
```

- 1) в зависимости от того сколько тип `int` занимает на пользовательской платформе. на моей он занимает 4 байта. стандарт C++ регламентирует, что размер `int` не может быть меньше 2 байт.
- 2) тут всё так же зависит от платформы. в моём случае - на 4 байта.
- 3) не совсем понял вопроса. расстояние между соседними элементами всегда 4 байта. если мы хотим узнать размер от элемента n до $(n + 3)$, то это $4 * 3 = 12$ байт.
- 4) `sizeof(arr) = 40`. 10 элементов массива по 4 байта каждый.
- 5) `iint *ptr_to_element = (ptr_to_begin + n)`.

Задание 2. Обработайте массив целых чисел `a[]` (выделение памяти и заполнение массива может быть выполнено на языке высокого уровня).

3	Найти сумму элементов массива
---	-------------------------------

task.s:

```
.globl sum
sum:
pushq %rbp
sub $16, %rsp
pushq %rsi
pushq %rdi

xorq %r12, %r12
xorq %r13, %r13
xorq %r14, %r14
xorq %r15, %r15

movq %rdi, %r12
movq %rsi, %r13
```

```

begin_iteration:
    cmp %r13, %r14
    jge end_loop

    movq (%r12, %r14, 4), %rsi
    addq %rsi, %r15

    incq %r14
    jmp begin_iteration
end_loop:

    movq %r15, %rax

    popq %rdi
    popq %rsi
    add $16, %rsp
    popq %rbp
    ret

```

main.cpp:

```

#include <iostream>

using std::cout;
using std::cin;
using std::endl;

#ifdef __cplusplus
extern "C" int sum(int* array, int numberOfElements);
#endif

static void printArray(int* array, int numberOfElements);
static int* generateArray(int numberOfElements);

int main(void) {
    int number;
    cout << "Enter the number of elements: ";
    cin >> number;
    if (cin.fail()) {
        cin.ignore();
        cin.clear();
        cout << "Error: wrong input parameter" << endl;
        return 0;
    }
}

```

```

}

if (number < 0) {
    cout << "Error: parameter 'number' must be positive" << endl;
    return 0;
}

int* array = generateArray(number);
printArray(array, number);
cout << endl;
int sum_of_elements = sum(array, number);
cout << endl << "Sum of elements = " << sum_of_elements << endl;

delete[] array;
return 0;
}

static int* generateArray(int numberOfElements) {
    int* generatedArray = new int[numberOfElements];
    if (generatedArray != nullptr) {
        for (int i = 0; i < numberOfElements; i++)
            generatedArray[i] = rand() % 15 + 4;
    }

    return generatedArray;
}

static void printArray(int* arr, int numberOfElements) {
    for (int i = 0; i < numberOfElements; i++)
        cout << arr[i] << " ";
    cout << endl;
}

```

```

kdator@kdator:~/study$ ./main
Enter the number of elements: 6
17 5 16 14 12 14

Sum of elements = 78

```

Задание 3. Создайте, используя язык высокого уровня, массив $\beta[]$ из структур, содержащих целый ключ и вещественное значение.

Определите размер массива и расположение элементов (используя возможности языка высокого уровня или IDE).

1. Каков размер структуры (в байтах)?
2. Каковы размеры полей структуры (в байтах)?
3. Насколько отличаются адреса соседних полей структуры (в байтах)?
4. Каков общий размер массива (в байтах)?
5. Как получить адрес поля элемента массива, зная номер элемента, адрес начала массива и имя поля?

```
#include <iostream>
#include <cstdint>

using std::cout;
using std::cin;
using std::endl;

struct int_float_map {
    int key;
    float value;
};

typedef int_float_map IntFloatMap;

int main(void) {
    IntFloatMap arr[5];

    cout << "An element size: " << sizeof(IntFloatMap) << endl;
    cout << "Int size: " << sizeof(((IntFloatMap *)0)->key)
        << ", Float size: " << sizeof(((IntFloatMap *)0)->value) << endl;
    cout << "Gap between elements: " << offsetof(IntFloatMap, value) -
        offsetof(IntFloatMap, key) << endl;
    cout << "The array size: " << sizeof(arr) << endl;
    return 0;
}
```

```
kdator@kdator:~/study$ ./main
An element size: 8
Int size: 4, Float size: 4
Gap between elements: 4
The array size: 40
```

- 1) 8.
- 2) каждый по 4.
- 3) 4.
- 4) `number_of_elements * 8`. у меня 40.
- 5) `(array_start + number_of_element + offsetof(struct_type, struct_field))`.

Задание 4. Реализуйте задание 2 для значений тех элементов массива $\beta[]$, ключ которых равен заданному числу k . Если таких нет, вывести корректное сообщение об этом.

task.s:

```
.data
STR:
    .string "key_sum: Elements not found\n"

.text
.globl key_sum
key_sum:
    pushq %rsi
    pushq %rdi
    pushq %rbx
    ;; temp_array
    xorq %r12, %r12
    ;; temp_size
    xorq %r13, %r13
    ;; i
    xorq %r14, %r14
    ;; res
    xorq %r15, %r15
    ;; array
    movq %rdi, %r12
    ;; numberOfElements
    movq %rsi, %r13

begin_iteration:
    cmp %r13, %r14
    jge end_loop

    ;; &arr[i]
    leaq (%r12, %r14, 8), %rsi
```

```

    //; if (a[i] == key)
    cmpl (%rsi), %edx
    jne false
    fld 4(%rsi)
    fldz
    faddp
    fstpl (%rbx)
false:

    incq %r14
    jmp begin_iteration
end_loop:

    movq %rbx, %rcx
    movd %rcx, %xmm0
    cmpl $0, %eax
    jne false2
    // Print call
    leaq STR(%rip), %rdi
    movb $0, %al
    callq printf
    xor %rax, %rax
    pxor %xmm0, %xmm0
false2:

    popq %rbx
    popq %rdi
    popq %rsi
    popq %rbp
    ret

```

new.cpp:

```

#include <iostream>

using std::cout;
using std::cin;
using std::endl;
using std::ios;

struct int_float_map
{
    int key;

```

```

    float value;
};

typedef int_float_map IntFloatMap;

static void printArray(IntFloatMap* array, int numberOfElements);
IntFloatMap* generateArray(int numberOfElements);

#ifdef __cplusplus
extern "C" float keySum(IntFloatMap* array, int numberOfElements, int
key);
#endif

int main(void) {
    srand(time(0));
    cout.setf(ios::fixed);
    cout.precision(5);

    int n = 10;
    double check_sum = 0;
    IntFloatMap* array = generateArray(n);
    printArray(array, n);

    float sum_of_elements = keySum(array, n, 1);

    for (int i = 0; i < n; i++) {
        if (i % 3 == 0) {
            check_sum += array[i].value;
        }
    }

    cout << "\nSum of elements = " << sum_of_elements
        << "\nCheck sum of elements = " << check_sum << '\n';
    return 0;
}

IntFloatMap* generateArray(int numberOfElements) {

    IntFloatMap* generated_array = new IntFloatMap[numberOfElements];
    if (generated_array != nullptr) {
        for (int i = 0; i < numberOfElements; i++) {
            if (i % 3 == 0) {
                generated_array[i].key = 1;
            }
        }
    }
}

```



```

    } else {
        generated_array[i].key = 2;
    }
    generated_array[i].value = (float)(rand()) / RAND_MAX;
}
}
return generated_array;
}

static void printArray(IntFloatMap* array, int numberOfElements) {
    for (int i = 0; i < numberOfElements; i++)
        cout << "key: " << array[i].key << " value: " << array[i].value <<
endl;
    cout << endl;
}

```

```

kdator@kdator:~/study$ ./new
key: 1 value: 0.58152
key: 2 value: 0.07741
key: 2 value: 0.58976
key: 1 value: 0.57846
key: 2 value: 0.08214
key: 2 value: 0.15396
key: 1 value: 0.14645
key: 2 value: 0.36665
key: 2 value: 0.24173
key: 1 value: 0.39169

Sum of elements = 1.69812
Check sum of elements = 1.69812

```

Задание 5. Опишите вставку либо функцию, инициализирующую массив заданной длины N первыми N элементами последовательности из задания 3 лабораторной работы №6.

fib.s:

```

.globl fibonacci
fibonacci:
    // загружаем данные со стека.
    pushq %rbp
    sub $16, %rsp
    pushq %rsi
    pushq %rdi

```

```
pushq %rbx
pushq %rax

// зануляем регистры.
xorq %r12, %r12
xorq %r13, %r13
xorq %r14, %r14

// параметры функции.
movq %rdi, %r12
movq %rsi, %r13

// внутренние переменные.
movq $1, %rsi
movq $1, %rdi
movq $1, %rbx

begin_iteration:
cmp %r13, %r14
jge end_loop

movq %rbx, %rdi

movq %rdi, %rax
addq %rsi, %rax
movq %rax, %rbx

movq %rdi, %rsi

movl %esi, (%r12, %r14, 4)

incq %r14
jmp begin_iteration
end_loop:

popq %rax
popq %rbx
popq %rdi
popq %rsi
add $16, %rsp
popq %rbp
ret
```

fib.cpp:

```
#include <iostream>

using std::cout;
using std::cin;
using std::endl;

#ifdef __cplusplus
extern "C" void fibonacci(int* array, int numberOfElements);
#endif

static void printArray(int* array, int numberOfElements);

int main(void) {
    int number;
    cout << "Enter the number of elements: ";
    cin >> number;
    if (cin.fail()) {
        cin.ignore();
        cin.clear();
        cout << "Error: wrong input parameter" << endl;
        return 0;
    }

    int array[number];
    fibonacci(array, number);
    printArray(array, number);

    return 0;
}

static void printArray(int* array, int numberOfElements) {
    for (int i = 0; i < numberOfElements; i++)
        cout << array[i] << " ";
    cout << endl;
}
```

```
Enter the number of elements: 17
1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584
```