

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

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

Код 1

```
#include <iostream>

#include <ctime>

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#define N 10

int main()

{

    srand(time(0));

    int *Array = new int [N];

    for(int i = 0; i < N; i++)

    {

        Array[i] = rand() % N;

        // std::cout << " " << Array[i];

        printf(" %p", &Array[i]);

    }

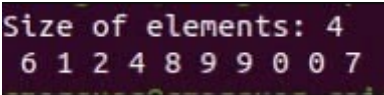
    std::cout << std::endl;

    delete [] Array;

    return (0);

}
```

Результат:



```
Size of elements: 4
6 1 2 4 8 9 9 0 0 7
```

Код 2

```
#include <iostream>

#include <ctime>

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

int main()

{

    srand(time(0));

    int N = 10;

    int *Array = new int [N];

    for(int i = 0; i < N; i++)

    {

        Array[i] = rand() % N;

        std::cout << " " << Array[i];

    }

    std::cout << std::endl;

    int Max = 0;

    asm (

        "movl $1, %%eax\n\t"

        "movl %[Array], %%ebx\n\t"

        "movl (%%ebx), %%ebx\n\t"

        "Start:\n\t"

        "cmpl %[N], %%eax\n\t"

        "je Exit\n\t"

        "movl %[Array], %%ecx\n\t"
```

```
"leal (%%ecx, %%eax, 4), %%ecx\n\t"
```

```
"movl (%%ecx), %%ecx\n\t"
```

```
"cmpl %%ebx, %%ecx\n\t"
```

```
"jc lable\n\t"
```

```
"movl %%ecx, %%ebx\n\t"
```

```
"lable:\n\t"
```

```
"incl %%eax\n\t"
```

```
"jmp Start\n\t"
```

```
"Exit:\n\t"
```

```
"movl %%ebx, %[Max]\n\t"
```

```
: [Max]" + m" (Max)
```

```
: [Array]" m" (Array), [N]" m" (N)
```

```
: "cc", "%eax", "%ebx", "%ecx"
```

```
);
```

```
std::cout << "max: " << Max << std::endl;
```

```
delete [] Array;
```

```
return (0);
```

```
}
```

Результат:

```
4 0 7 1 1 9 7 1 3 0
max: 7
Ошибка сегментирования (стек памяти сброшен на диск)
smarguer@smarguer-cpliscine21:~$ g++ lab7.c && ./a.out
6 6 2 5 2 4 1 2 2 3
max: 7
Ошибка сегментирования (стек памяти сброшен на диск)
smarguer@smarguer-cpliscine21:~$ g++ lab7.c && ./a.out
0 9 6 8 4 8 9 4 8 7
max: 7
Ошибка сегментирования (стек памяти сброшен на диск)
smarguer@smarguer-cpliscine21:~$ g++ lab7.c && ./a.out
7 7 5 1 4 7 2 6 6 1
max: 9
Ошибка сегментирования (стек памяти сброшен на диск)
smarguer@smarguer-cpliscine21:~$ g++ lab7.c && ./a.out
7 1 9 1 5 5 6 5 8 2
max: 9
```

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).

Код 3:

```
#include <iostream>
```

```
#include <ctime>
```

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include <time.h>
```

```
#define N 10
```

```
using namespace std;
```

```
struct s_struct
```

```
{
```

```
int key;
```

```
double value;
```

```
};
```

```
int main()
```

```
{
```

```
srand(1234);
```

```
struct s_struct *Array = new struct s_struct[N];
```

```
for (int i = 0 ; i < N ; i++)
```

```
{
```

```
Array[i].key = rand() % (53);
```

```
Array[i].value = rand() % (32) / 1.324;
```

```
printf("Key = %d, Val = %.3f\n", Array[i].key, Array[i].value);
```

```
}
```

```
cout << " \nразмер структуры: " << sizeof(Array[0]) << "\nразмер ключа: " << sizeof(Array[0].key) << "\nразмер
```

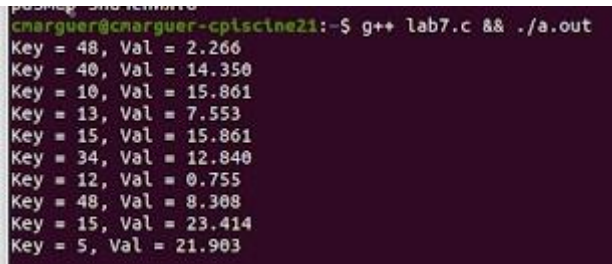
```
значения:" « sizeof(Array[0].value) « endl;
```

```
delete [] Array;
```

```
return (0);
```

```
}
```

Результат:



```
cmarguer@carguer-cpiscine21:~$ g++ lab7.c && ./a.out
Key = 48, Val = 2.266
Key = 40, Val = 14.350
Key = 10, Val = 15.861
Key = 13, Val = 7.553
Key = 15, Val = 15.861
Key = 34, Val = 12.840
Key = 12, Val = 0.755
Key = 48, Val = 8.308
Key = 15, Val = 23.414
Key = 5, Val = 21.903
```

начала массива и имя поля :

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.

Код 4:

```
#include <iostream>
```

```
#include <ctime>
```

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include <time.h>
```

```
using namespace std;
```

```
struct Array
```

```
{
```

```
int k;
```

```
int value;
```

```
};
```

```
int main()
```

```
{
```

```
srand(time(0));
```

```
int N = 10;
```

```

Array *Array = new struct Array [N];

for(int i = 0; i < N; i++)

{

Array[i].value = 1+rand() % N;

Array[i].k=1+rand()%N/2;

std::cout << " " <<Array[i].k<<"->"<<Array[i].value;

}

std::cout << std::endl;

int Max = 0;

int key =4;

asm (

"movl $1, %%eax\n\t"

"movl $0, %%ebx\n\t"

"movl $0, %%ecx\n\t"

"Start:\n\t"

"cml $N, %%eax\n\t"

"je Exit\n\t"

"movl %[Array], %%ecx\n\t"

"leal (%%ecx, %%eax, 8), %%ecx\n\t"

"movl (%%ecx),%%ecx\n\t"

"cml %[key],%%ecx\n\t"

"jne notkey\n\t"

"movl %[Array], %%ecx\n\t"

"leal 4(%%ecx, %%eax, 8), %%ecx\n\t"

"movl (%%ecx),%%ecx\n\t"

"cml %%ebx, %%ecx\n\t"

"jc lable\n\t"

```

```
"movl %%ecx, %%ebx\n\t"
```

```
"labe:\n\t"
```

```
"notkey: \n\t"
```

```
"incl %%eax\n\t"
```

```
"jmp Start\n\t"
```

```
"Exit:\n\t"
```

```
"movl %%ebx, %[Max]\n\t"
```

```
: [Max]" + m" (Max)
```

```
: [Array]"m" (Array), [N]"m"(N), [key]"m"(key)
```

```
: "cc", "%eax", "%ebx", "%ecx"
```

```
);
```

```
cout << endl;
```

```
if (Max == 0)
```

```
cout << "Array not have elements for key: " << key ;
```

```
else
```

```
cout << "max: " << Max;
```

```
cout << endl;
```

```
delete [] Array;
```

```
// delete [] intArray;
```

```
return 0;
```

```
}
```

Результат:

```
3->1 1->8 2->2 3->4 1->5 4->1 2->2 5->9 1->4 1->3
```

```
5->7 4->4 1->9 3->3 5->4 3->10 2->4 5->1 1->10 1->2
```

```
Array not have elements for key: 0
```